INVESTIGATIONS INTO EFFECTIVELY MOVING PEOPLE AND GOODS

A Dissertation
Presented to
The Academic Faculty

By

Idil Arsik

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the Industrial Engineering
School of H. Milton Stewart School of Industrial and System Engineering

Georgia Institute of Technology

August 2020

INVESTIGATIONS INTO EFFECTIVELY MOVING PEOPLE AND GOODS

Approved by:

Dr. Martin W.P. Savelsbergh (Thesis advisor)
Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Alan L. Erera
Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Natashia Boland
Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Alejandro Toriello
Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Mauricio G.C. Resende
Industrial and Systems Engineering
University of Washington and Amazon

Date Approved: May 14, 2020

To my parents Hermine & Arsin.

ACKNOWLEDGEMENTS

Getting Ph.D. at Georgia Tech was one of the most challenging yet one of the most rewarding experiences in my life. It was not a smooth journey. But it thought me how to be patient, how to grow from hardship and how to question and think critically as a researcher. Along the way, there were many brilliant people who influenced me and supported me academically and personally. I would like to acknowledge and thank them.

First and foremost, I want to express my sincere gratitude to my advisor, Dr. Martin Savelsbergh for giving me the opportunity to work with him and learn from his experience. I am indebted to him for his support and guidance when I needed them the most. I will always be thankful to him for being generous with his time and sharing his knowledge in the most humble and patient way possible. I would also like to thank my committee members: Dr. Natashia Boland, Dr. Alan Erera, Dr. Mauricio Resende and Dr. Alejandro Toriello for devoting their time to serve in my thesis committee and for sharing their invaluable comments and questions. I would also like to extend my sincere thanks to Dr. Setareh Mardan and Dr. Timothy L. Jacobs for their support.

I have been very fortunate to meet a bunch of wonderful people whose friendship I will always cherish. Thank you very much to all of you: To my Turkish ISyE friends: İlke, Tuğçe, Ezgi, Şeyma, Beste, Burak, Murat, Bahar, Gizem. To my Chilean friends: Alfredo, Matias, Berni, Daniela, Ramon, Sebastián. To Reem, Tony, Mina, Timur, Adrian, Andrew, Damián, Kelly, Luke, Asteroid, Nayeon, Kirthana and to the entire ISyE family. To non-ISyE friends: Fatma, Micheal, Didem, Can,Gözde, Bige, Esma.

A special thanks goes to my better half, Enis. It was your patience and understanding that helped us survive this journey. Thank you for always being there for me, being the voice of reason I sometimes could not hear and believing in me when I failed to do so.

Lastly, I want to thank my parents Hermine and Arsin, and my big brother Arda. Without your endless support and unconditional love, this thesis would not be possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

In this thesis, we investigate practical methods to move people and goods effectively on the road network. In the first part of this thesis, we focus on route guidance for the movement of people while in the second part we focus on the movement of goods by investigating the two main aspects of service network design: *flow* and *resource* planning.

In Chapter 2, we introduce a centralized proactive route guidance approach motivated by the anticipated introduction of autonomous vehicles which, with full adoption, can create an environment in which a specific origin-destination path can be assigned to each self-driving vehicle and the vehicle follows the assigned path. Our approach integrates a system perspective, i.e., minimizing congestion, and a user perspective, i.e., minimizing inconvenience. As a design choice, we only solve linear program which is more likely to scale well and be of practical use. The linear structure of our models allows us to derive theoretical properties. In particular, we show that for the problem of minimizing maximum arc utilization,which is used as a measure of congestion in a road network, results analogous to those well-known for the maximum flow problem, e.g., the max flow-min cut theorem, can be derived.

In Chapter 3, we focus on cost-effective routing of commodities on a line-haul network from its origin to its destination while meeting tight service requirements, satisfying operational constraints and minimizing transportation costs. We introduce a marginal cost path-based greedy heuristic that works with a partially time-expanded network to solve large scale real-life instances found in practice. Our approach involves two consolidation improvement heuristics and novel use of iterative refinement within the greedy heuristic to obtain a continuous-time feasible service network design.

In Chapter 4, we analyze the value of outsourcing transportation for different negotiated prices with contractors and, while doing so, explicitly account for driver considerations. We introduce a depth-first search algorithm to generate a set of time-feasible cycles of chosen

length in terms of the number of dispatches that covers a set of planned dispatches in a given load plan, where dispatches can be connected by empty travel. When generating cycles, we respect the company specific rules and hours-of-service regulations that ensure road safety and prevent fatigue related accidents. We solve an integer programming model that identifies a subset of company cycles (and contractor cycles and one-way moves if the outsourcing option is available) that maximizes the cost savings over the (unrealistic) scenario in which company drivers perform a one-way move and return empty. When a company only performs out-and-back cycles, as is current practice at SF Express, one of the largest Chinese small parcel carriers, we efficiently choose the set of cycles by solving bipartite matching problem for each out-and-back lane pair separately.

CHAPTER 1

INTRODUCTION

The physical movement of people and goods from one place to another may seem to be one of the more mundane activities and one that can be taken for granted in our daily life. However, a slow-down or interruption in any of these movements can have a cascading negative impact on economic activities and social life.

One of the main causes of a slow-downs, or even interruptions, of movements is congestion on the road network which occurs when the existing road infrastructure falls short to serve the traffic load. Especially in urban areas, rush-hour traffic and the resulting delays and slow-downs are common due to high volume of people commuting around the same hours of the day to start and end their work shifts. Combined with high traffic load, the lack of communication and coordination among drivers exacerbate the negative impacts of the congestion on people, businesses and environment. The 2019 Urban Mobility Report shows that, in 2017, the average auto commuter in the U.S. spent an extra 54 hours travelling and wasted 21 gallons of fuel due to congestion [1]. According to the same report, the congestion cost of extra time and fuel in 494 U.S. urban areas was $179 billion of which 11% ($20 billion) was related to truck operations.

In addition to the traffic conditions, the effective movement of goods over long distances heavily depends on i) linking full truckload movements by short empty movements and ii) identifying consolidation opportunities for less-than truckload movements. The ongoing pandemic of coronavirus disease 2019 (COVID-19), which, as of May 2020, infected more than 3 million people worldwide [2] stressed the importance of innovative and efficient transportation and logistics activities to ensure *fast*, *safe* and *reliable* movement of goods both in normal and unprecedented times. To prevent the spread of the disease, many local authorities/governments worldwide have suspended public transit operations and have issued domestic and international travel restrictions. Thus, people started to rely more on online shopping for safe and timely delivery of essential food, goods and medical supplies. The role played by e-commerce providers and express parcel delivery services became more critical than ever. For instance, express parcel delivery company SF Express provided the delivery of essential supplies to Wuhan, the epicenter of the global outbreak [3]. To minimize human contact, JD Logistics, one of the largest e-commerce providers in China, has deployed its autonomous ground robots and drones for last-mile delivery of supplies to residents and hospitals in Wuhan [4].

In this thesis, we investigate practical methods to move people and goods effectively over a road network with the goal of scalability and realistic representation of real-life constraints, which are imperative in real-world settings. In the first part of this thesis, we focus on route guidance for the movement of people while in the second part we focus on the movement of goods by investigating the two main aspects of service network design: *flow* and *resource* planning for one of the largest Chinese small parcel carriers, SF Express.

In Chapter 2, we introduce a centralized proactive route guidance approach motivated by the anticipated introduction of autonomous vehicles which, with full adoption, can create an environment in which a specific origin-destination path can be assigned to each self-driving vehicle and in which that vehicle will follow the assigned path. The goal is to reduce or avoid congestion in the road network without causing an excessive increase in the length (or duration) of the paths traveled by individuals, when compared to the shortest (least-duration) paths between their origin and destination. Given a maximum level of travel inconvenience, the approach obtains a system-optimal distribution of traffic using a hierarchical approach. First, a linear programming model is used to minimize the maximum level of congestion which is measured by the arc utilization, i.e. the ratio of the number of vehicles entering an arc per unit time and its capacity. Second, for a given minimum maximum level of congestion, another linear programming model is used to minimize the weighted average experienced travel inconvenience subject to the constraint that the maximum congestion does not exceed the minimum possible value computed using the first model. Our extensive computational study shows that, in many settings, relatively small values of maximum level of travel inconvenience can lead to congestion avoidance.

In Chapter 3, we focus on cost-effective routing of commodities in a line-haul network from their origins to their destinations while meeting tight service requirements, satisfying operational constraints and minimizing transportation costs. We introduce a marginal cost path-based greedy heuristic that works with a partially time-expanded network to solve large scale real-life instances found in practice. Our approach involves two consolidation improvement heuristics and a iterative refinement heuristic algorithm inspired by [5], which iteratively eliminates infeasibilities introduced by the discretization of time and the representation of the timing of events. The output is a *load plan* which specifies the vehicle dispatches between terminals and the origin-destination route for each commodity (in terms of vehicle dispatches from each visited terminal).

In Chapter 4, we look into identifying time-feasible driver schedules to execute a load plan, i.e., the set of planned vehicle dispatches, so as to minimize the total transportation and operational costs (e.g. layover cost within driver routes) while respecting hours-of-service regulations and company rules. In addition to company operations, we consider the option of outsourcing transportation and assess its benefits for different price points. Our study can inform both long and short term resource planning decisions, especially in the presence of outsourcing options.

1.1 Contributions

Unlike previous studies in traffic assignment, which tend to focus on either the user or the system equilibrium, our approach integrates a system perspective, i.e., minimizing congestion, and a user perspective, i.e., minimizing inconvenience. The approach focuses on reducing, ideally avoiding, congestion while minimizing the experienced travel inconvenience. In addition, our design choice of only solving linear programs has significant computational advantages over previously proposed approaches and is more likely to scale well and be of practical use. The linear structure of our models also allows us to derive theoretical properties. In particular, we show that for the problem of minimizing maximum arc utilization, results analogous to those well-known for the maximum flow problem, e.g., the max flow-min cut theorem, can be derived. We note that the study in Chapter 2 is joint work with a group of researchers at the University of Brescia, who were primarily responsible for the computational aspects of the research.

In Chapter 3, we introduce a marginal cost path-based greedy heuristic for the solution of large-scale instances of the service network design problem, i.e., the routing of freight from its origin to its destination while minimizing total transportation costs in the transportation (line-haul) network. We also introduce the novel use of iterative time refinement within the greedy heuristic to obtain a continuous-time feasible solution of the service network design problem. Our algorithmic design choices are motivated by the need/desire to solve real-world instances for which it is impossible to rely on integer programming approaches (even with the most advances, fastest commercial solvers). In addition to its speed, the approach also allows the incorporation of many real-world complexities, which is necessary if optimization technology is to be used in practice.

To best of our knowledge, there are few, if any, studies in the literature that address driver considerations and outsourcing opportunities in service network design, even though these are critically important in real-world settings. Inspired by the operations of SF Express, we analyze the value of outsourcing transportation for different negotiated prices with contractors and, while doing so, explicitly account for driver considerations. We introduce a depth-first search algorithm to generate a set of time-feasible cycles of chosen length in terms of the number of dispatches that covers a set of planned dispatches in a given load plan, where dispatches can be connected by empty travel. When generating cycles, we respect the company specific rules and hours-of-service regulations that ensure road safety and prevent fatigue related accidents. We solve an integer programming model that identifies a subset of company cycles (and contractor cycles and one-way moves if the outsourcing option is available) that maximizes the cost savings over the (unrealistic) scenario in which company drivers perform one-way moves and return empty. When a company only performs out-and-back cycles, as is current practice at SF Express, we efficiently choose the set of cycles by solving bipartite matching problem for each out-and-back lane pair separately.

Both in Chapter 3 and Chapter 4, extensive computational studies demonstrate the need and benefit of incorporating realistic aspects of service network design and demonstrate the value of algorithmic choices in terms of both solution quality and speed.

Remark

In addition to the research presented in this dissertation, during my Ph.D. studies, I have also worked with Prof Pinar Keskinocak from the School of Industrial and Systems Engineering at Georgia Institute of Technology on two health-care related projects which resulted in two papers. In [6], we assessed the implementation of gait evaluation in neurological clinics with the goal of understanding the the relationship between gait and cognitive health outcomes over a 21-month period with more than 500 patients. In [7], we conducted a retrospective analysis using data from the TONIC trail ([8]) to validate alanine aminotransferase (ALT), a monitoring biomarker for change in liver histology. Our results suggest that ALT as a valid monitoring biomarker of histologic change over time in children with nonalcoholic steatohepatitis (NASH) and fibrosis.

CHAPTER 2
PROACTIVE ROUTE GUIDANCE TO AVOID CONGESTION

2.1 Introduction

The fraction of the population living in urban areas continues to grow. As a consequence, traffic in urban areas is increasing and the inability to significantly increase road network infrastructure is making the issue of traffic control and coordination more and more relevant and pressing. Congestion is a common phenomenon experienced in cities and towns around the world, and causes delays, stress, and pollution. The negative impact of congestion on the economy, the society, the environment, and on people's health is enormous. Government, industry, and private citizens are all interested in ways to reduce the negative externalities of transportation. Technology has always been an integral part of attempts to alleviate congestion, but recent and anticipated technological and automotive advances offer enormous and exciting new opportunities.

Traditionally, traffic information has been communicated to drivers via radio or by means of Variable Message Signs. The drawback of these systems is that the information being communicated is the same for all drivers and, as such, has only limited value in globally coordinating traffic. The most common modern in-vehicle device aimed at helping drivers guide a vehicle in a road network is a car navigation system based on a digitalized road network map and a global positioning system (GPS) aerial. The GPS aerial allows the vehicle to be localized on the map, and embedded optimization software allows the selection of the best route to the destination. Based on the available information on the status of the road network, the navigation system may provide an optimal route to the destination with respect to the user's preference, which can be the shortest path in terms of distance or travel time, or the least expensive path in terms of fuel consumption or, even, emissions produced. Challenges (and frustrations) occur when the road utilization on the route proposed by the navigation system exceeds its capacity and congestion occurs. In fact, and especially during peak hours, congestion often occurs because the paths of many vehicles traverse the same sections of the road network. Recently, navigation systems have been integrated with real-time traffic data acquisition systems that allow detection of traffic jams and/or road interruptions and offer the potential to reroute the drivers to different paths. Unfortunately, these systems typically do not (yet) consider the system-wide impact of the directions they provide to the drivers. The navigation devices, again, provide drivers with the same information and route guidance, which, in many cases, simply shifts the congestion to other parts of the road network.

The drawbacks of user-optimal paths, which result in a user equilibrium state of the traffic network, compared to a system-optimal set of user paths have long been investigated and understood (see, for example, [9]). It is also well-known that, with a system-optimal set of user paths, some users may end up being assigned to paths that are much longer, in distance or time, than the shortest possible path between their origin and destination. This unfairness, among others, results in users not following route guidance, especially when suggested routes deviate substantially from a user's preference and are (expected to take) much longer.

However, technological and automotive advances may change the situation favorably. The anticipated introduction of autonomous or self-driving vehicles may drastically alter the landscape. Massive adoption of self-driving vehicles will have several benefits in terms of congestion. First and foremost, because the safe separation distance between two self-driving vehicles will be much smaller, their introduction will implicitly alter the capacity of the road network. Secondly, drivers will get used to trusting their vehicle to get them from their origin to their destination and, as a consequence, will be more likely to accept a route that deviates from the preferred (shortest) route. The latter, forms the motivation and underpinning of our research. In particular, we focus on an environment in which a specific origin-destination path can be assigned to each (self-driving) vehicle and in which that vehicle will follow the assigned path.

We propose a centralized proactive route guidance approach that integrates a system perspective, focused on reducing, ideally avoiding, congestion, and a user perspective, focused on minimizing the experienced travel inconvenience. The goal is to reduce or avoid congestion without causing an excessive increase in the length (or duration) of the paths traveled by individuals, when compared to the shortest (least-duration) paths between their origin and destination.

Our starting point is a road network and an Origin-Destination (OD) matrix specifying the number of trips that are estimated to take place between each origin and each destination. The problem of forecasting traffic on a road network has been well studied (see, for example, [10]; [11]; [12], and more recently [13])) and traditionally has been modeled in four steps: trip generation, trip distribution, mode choice, and traffic assignment. In particular, the zonal interchange analysis of trip distribution provides the so called Origin-Destination (OD) matrix, that is the matrix that provides for each OD pair the number of trips with the same origin and destination. Starting from Merchant and [14], researchers have also studied a dynamic traffic assignment problem which presents additional challenges (see [15]; [16] and more recently [17]).

The time period of interest is the rush hour, which in large cities may last a few hours, and in which, as [12] points out, traffic often exhibits a steady-state behavior. We assume that the arcs of the road network are characterized by a capacity representing the maximum number of vehicles per time unit at which vehicles can enter the arc and experience no slowdown due to congestion effects. The proposed approach will assign paths to drivers so as to minimize congestion while not increasing their experienced travel inconvenience too much. A maximum level of travel inconvenience is ensured and a certain level of fairness is maintained by limiting the set of considered paths for each Origin-Destination pair to those whose relative difference with respect to the shortest (least-duration) path, called travel inconvenience, is below a given threshold.

An important feature of the approach, and a critical design choice, is that only linear programs are solved. To have any potential practical value, a route guidance approach has to be computationally efficient. Computational efficiency has prompted us to restrict ourselves to the use of linear programming models (even if that would mean sacrificing some accuracy in our modeling choices).

The distribution of traffic is evaluated by two measures: the minimum maximum arc utilization in the network (a system perspective) and the weighted average experienced travel inconvenience (a user perspective). Arc utilization, i.e., the ratio of the number of vehicles entering an arc per time unit and its capacity, is used as an arc congestion measure. The weighted average experienced travel inconvenience averages the experienced travel inconvenience over all possible paths weighted by the number of vehicles that enter the path per time unit. The weighted average experienced travel inconvenience is minimized under the constraint that the minimum maximum arc utilization does not exceed a given limit: the minimum maximum arc utilization achievable if it is greater than one, or one if it is smaller than one (by definition no slowdown due to congestion effects occurs when the minimum maximum arc utilization is one).

The linear structure of the models allows us to derive theoretical properties. We will show in particular that for the problem of minimizing the maximum arc utilization, results analogous to those well known for the maximum flow problem, such as the max flow-min cut theorem, hold.

An extensive and comprehensive computational study demonstrates that in many settings relatively small values of the maximum travel inconvenience lead to a minimum maximum arc utilization less than or equal to one, i.e., avoidance of congestion effects. The computational tests are carried out using randomly generated benchmark instances that represent characteristic features of the most common circular-shaped city road networks. As mentioned before, our assumption is that (self-driving) vehicles follow their assigned path, i.e., a 100 % compliance rate. Our investigation shows, among others, that compliance is critical; it becomes much more difficult to reduce or avoid congestion when the recommended paths are not followed.

Finding a system-optimal traffic distribution that ensures a certain level of fairness is also considered by [18]. They also limit the set of paths for an OD pair to limit the travel inconvenience. However, they model the arc travel time as a function of the number of vehicles on that arc (using a widely adopted non-linear increasing function). The model assigns paths to users with the objective of minimizing the total user travel time. The model is non-linear and a column generation solution method is proposed and tested on real road networks. The approach we propose is substantially different as it seeks to avoid congestion by minimizing the maximum arc utilization. We find the minimum congestion level first and then minimize the user travel inconvenience.

A few other related papers have been published. In [19], a static multi-objective approach seeking to minimize the mean travel time cost and the travel time variance is presented, while in [20] a centralized path assignment model is proposed in which the objective is to minimize the geometric path duration mean for all drivers in the system. Centrally controlled traffic systems are often encountered in large warehouses where automated guided vehicles have to be routed (see [21]). In [22], an integer programming model is proposed which seeks to minimize the weighted path length of a set of automated guided vehicles.

The remainder of this chapter is organized as follows. In Section 2.2, the proactive route guidance setting considered is formally introduced, the path-based linear programming models comprising our proactive route guidance approach are given, as well as an algorithm to generate the sets of eligible paths, and supporting theoretical results are presented. In Section 2.3, we discuss the generation of road networks on which the approach is tested, discuss detailed results for a specific instance, and average results for the complete set of instances. Finally, some conclusions are drawn in Section 2.4.

2.2 Proactive route guidance

The basic idea of the system-optimal approach we propose is to assign paths other than the shortest (least-duration) path to vehicles in order to reduce, and possibly avoid, congestion in the road network, but to do so in a way that minimizes the inconvenience experienced by the drivers, in part by only considering alternative paths whose relative increase with respect to the shortest path is within a pre-defined limit. That is, we impose a limit on user travel inconvenience, in terms of the maximum allowed increase relative to the shortest path, which we denote by τ and refer to as the *maximum travel inconvenience*.

More specifically, for each OD pair, we generate a set of eligible paths, which are those Origin-Destination paths with a travel inconvenience that is no more than the specified limit τ. Furthermore, we assume that, for the time period of interest, the demand associated with an OD pair is specified in terms of the number of vehicles entering the network at the origin per time unit. The basic premise of the approach is that the congestion of the road network depends on the utilization of its arcs, where the utilization of an arc is the ratio of the number of vehicles entering an arc per time unit and the maximum number of vehicles per time unit at which vehicles can enter the arc and experience no slowdown due to congestion effects. When the flow per time unit on an arc exceeds its capacity, i.e. the arc utilization is greater than 1, the arc is said to be congested. We say that the road network is congested if at least one of its arcs is congested.

Given a maximum travel inconvenience, and, thus, a set of eligible paths for each OD pair, the approach obtains a system- optimal distribution of traffic using a hierarchical approach. First, a linear programming model is used to minimize the maximum arc utilization, i.e., the level of congestion. If the minimum maximum arc utilization exceeds one, then congestion in the road network is unavoidable for the imposed maximum travel inconvenience; to reduce the level of congestion even further, longer, less convenient paths need to be allowed. On the other hand, when the minimum maximum arc utilization is less than or equal to one, congestion can be avoided with the imposed maximum travel inconvenience. In fact, there is no reason to seek a minimum maximum arc utilization below one, because it can only come at the expense of unnecessary user travel inconvenience. Second, for a given minimum maximum arc utilization, another linear programming model is used to minimize the weighted average experienced travel inconvenience subject to the constraint that the maximum arc utilization does not exceed the minimum possible value computed using the previous model (or one if this value is less than one).

Our approach focuses on a system that exhibits a steady-state behavior and seeks to determine the minimum maximum travel inconvenience that allows elimination of congestion in the system by proactive route guidance, if such a maximum travel inconvenience exists. It is possible, of course, that the optimization reveals that such a maximum travel inconvenience does not exist, due to a combination of network infrastructure characteristics, e.g., a small number of bridges and/or tunnels connecting different parts of the city, and demand characteristics, e.g., extremely high demand. In such situations, more elaborate linear programming models are needed. In Section 4, we briefly discuss this possible extension.

Next, we present the two path-based linear programming models used in the approach, followed by the algorithm for generating eligible paths and supporting theoretical results.

2.2.1 Optimization Models

We consider a directed network *G = (V, A)*, where *V* represents the set of vertices and $A \subseteq V \times V$ represents the set of arcs. Each arc $(i, j) \in A$ represents a road segment on which vehicles can travel. The length of arc $(i, j) \in A$, depending on the user's preferences, may represent the traveling time or the length in space, and it will be denoted by l_{ij}. The capacity of each arc $(i, j) \in A$ is denoted by u_{ij} and represents the maximum rate (number of vehicles per time unit) that can enter an arc (i, j) at the average speed. We consider a set C of OD pairs. An OD pair $c \in C$ is described by its origin $O_c \in V$, its destination

$D_c \in V$ and a flow rate (number of vehicles per time unit) d_c, representing the rate of vehicles traveling from O_c to D_c. We let $D = \sum c \in Cd_c$ be the total flow rate in the network. Let τ be the maximum travel inconvenience. A set of paths K_c^τ is associated with each OD pair c and contains all paths that are not longer than $(1+\tau)SP_c$, where SP_c is the shortest path from the origin O_c to the destination D_c . The travel inconvenience τ_c^k associated with path $k \in K_c^\tau$, $c \in C$, is defined as the relative increase of path k with respect to the shortest (least-duration) path SP_c. Denoting by l_c^k the length of path k for OD pair c, we have $\tau_c^k = \frac{l_c^k - SP_c}{SP_c}$.

The optimization problem formulations use an indicator a_{ij}^{kc}, which takes value 1 if path $k \in K_c^\tau$ contains arc $(i,j) \in A$, and 0 otherwise. The critical parameter ρ_τ^* represents the minimum possible value of the maximum arc utilization. This value is computed by a linear programming model, referred to as the *congestion model* and defined below. The decision variables y_c^k represent the flow rate (number of vehicles per time unit) of OD pair $c \in C$ routed on path $k \in K_c^\tau$. The auxiliary variables x_{ij} and ρ represent the total flow rate (number of vehicles per time unit) traveling on arc $(i,j) \in A$ and an upper bound on the maximum arc utilization, respectively. The *inconvenience model* is a linear programming model seeking to minimize the weighted average experienced travel inconvenience over all OD pairs, defined as $\frac{1}{D} \sum\limits_{c \in C} \sum\limits_{k \in K_c^\tau} \tau_c^k y_c^k$. A summary of the sets, parameters and variables is given in Table 2.1.

Table 2.1: Notation for route guidance

Sets	
V	Set of vertices
A	Set of arcs
C	Set of OD pairs
K_c^τ	Set of eligible paths for $c \in C$ with maximum travel inconvenience τ
Parameters	
u_{ij}	Capacity of arc $(i,j) \in A$
a_{ij}^{kc}	1 if path $k \in K_c^\tau$ contains arc $(i,j) \in A$, 0 otherwise
SP_c	Length of shortest path for OD pair $c \in C$
l_c^k	Length of path $k \in K_c^\tau$
τ_c^k	Relative length increment of path $k \in K_c^\tau$,with respect to SP_c : $\tau_c^k = \frac{l_c^k - SP_c}{SP_c}$
d_c	Flow rate for OD pair $c \in C$
D	Total flow rate over all OD pairs : $D = \sum\limits_{c \in C} d_c$
α	Compliance rate
Decision Variables	
y_c^k	Flow rate of OD pair $c \in C$ routed on path $k \in K_c^\tau$
Auxillary Variables	
x_{ij}	Total flow rate entering arc $(i,j) \in A$: $x_{ij} = \sum\limits_{c \in C} \sum\limits_{k \in K_c^\tau} a_{ij}^{kc} y_c^k$
ρ	Upper bound on arc utilization : $\rho \geq \frac{x_{ij}}{u_{ij}}$ $\forall (i,j) \in A$
f_{ij}^c	Flow rate of OD pair $c \in C$ traversing arc $(i,j) \in A$

The inconvenience model

$$\min \quad \frac{1}{D} \sum_{c \in C} \sum_{k \in K_c^\tau} \tau_c^k y_c^k$$

subject to

$$\rho \leq max(1, \rho_\tau^*) \tag{2.1}$$

$$\frac{x_{ij}}{u_{ij}} \leq \rho \qquad \forall (i,j) \in A \tag{2.2}$$

$$x_{ij} = \sum_{c \in C} \sum_{k \in K_c^\tau} a_{ij}^{kc} y_c^k \qquad \forall (ij) \in A \tag{2.3}$$

$$d_c = \sum_{k \in K_c^\tau} y_c^k \qquad \forall c \in C \tag{2.4}$$

$$\rho \geq 0 \tag{2.5}$$

$$x_{ij} \geq 0 \qquad \forall (ij) \in A \tag{2.6}$$

$$y_c^k \geq 0 \qquad \forall c \in C, \qquad k \in K_c^\tau \tag{2.7}$$

Constraint (2.1) guarantees that the maximum arc utilization is minimized in case congestion is unavoidable ($\rho^*_\tau > 1$), and ensures that the flow rate on any arc is less than or equal to the arc's capacity, otherwise ($\rho^*_\tau \leq 1$). Constraints (2.2) bound the maximum arc utilization, i.e., $\max\limits_{(i,j)\in A} \frac{x_{ij}}{u_{ij}} \leq \tau$ and constraints (2.3) set the flow rate on arc (i, j). Constraints (2.4) ensure that the required flow rate d_c of an OD pair $c \in C$ is routed on (a subset of) its eligible paths. Finally, constraints (2.5) - (2.7) define the domains of the decision variables. Although the natural domain of variables y^c_k and x_{ij} is the set of non- negative integers, it is reasonable to relax it to the set of real numbers as long as the values d_c are large. Note that $\frac{y^k_c}{d_c}$ can be interpreted as the fraction of the flow rate d_c that is routed on path $k \in K^\tau_c$.

The minimum maximum arc utilization ρ^*_τ required in constraint (2.1) is computed with the following linear programming model.

The congestion model

$$
\begin{aligned}
\rho^*_\tau \equiv \text{min} \quad & \rho \\
\text{subject to} \qquad & (2.2) - (2.7)
\end{aligned}
$$

The congestion model focuses purely on congestion, i.e., on minimizing the maximum arc utilization, without taking into account the impact on the drivers. On the other hand, the inconvenience model without constraints (2.1) - (2.2) focuses purely on user travel inconvenience without taking into account the impact on congestion. For this reason, a hierarchical approached is used. We first focus on congestion and then focus on user travel inconvenience. The following remarks provide basic relations linking the maximum travel inconvenience τ, the path set K^τ_c, and the minimum value of the maximum arc utilization ρ^*_τ.

Remark 1. *Let $K^{\tau_1}_c$, $K^{\tau_2}_c$ be the sets of eligible paths for OD pair $c \in C$ for τ_1 and τ_2, respectively. If $\tau_1 \leq \tau_2$, then $K^{\tau_1}_c \subseteq K^{\tau_2}_c$.*

Remark 2. *Let $K_c^{\tau_1}$, $K_c^{\tau_2}$ be the sets of eligible paths for OD pair $c \in C$ for τ_1 and τ_2, respectively. If $K_c^{\tau_1} = K_c^{\tau_2}$ $\forall c \in C$, then the minimum maximum arc utilization is the same for the two settings.*

Remark 3. *Let $\rho^*_{\tau_1}, \rho^*_{\tau_2}$ be the optimum values of the congestion model when the maximum travel inconvenience is τ_1 and τ_2 respectively. If $\tau_1 \leq \tau_2$, then $\rho^*_{\tau_1} \geq \rho^*_{\tau_2}$.*

The models can be modified to take into account that not all vehicles may follow the recommended path. We define the compliance rate to be the fraction α of vehicles following the recommended path and assume that 1-α vehicles, instead, choose the shortest (least-duration) path. For sake of simplicity, we assume that the compliance rate is identical for all OD pairs. The compliance rate is taken into account by adding to each model the constraints.

$$y_c^{SP} \geq (1-\alpha)d_c \quad c \in C \tag{2.8}$$

These constraints impose that at least a fraction 1-α of the demand is routed along its shortest path. This fraction represents those drivers acting selfishly.

2.2.2 Generation of the eligible paths

To the best of our knowledge, whereas the problem of finding the first K shortest paths is well known (see [23] and [24]), no algorithm has been published for the computation of all paths between two vertices whose length does not increase the shortest path length by a given percentage. Thus, in order to compute the set K_c^{τ} , an ad hoc algorithm has been developed. The algorithm relies on depth-first search and consists of a main body *ConstructingEligiblePaths* (sketched in Algorithm 1) and a recursive routine *ScanVertex* (sketched in Algorithm 2).

Inputs for algorithm *ConstructingEligiblePaths* are the network *(V, A)*, the set of OD pairs *C*, and the maximum travel inconvenience τ. The algorithm first computes the shortest path matrix (the shortest path for each pair of vertices) and labels all vertices as inactive. Labels will change from inactive to active and back during computation. Next, for each OD pair $c \in C$, initializations are made and a depth-first search to find the set K_c^τ is performed by means of the recursive routine *ScanVertex*.

Routine *ScanVertex* operates as follows. The input vertex i is labeled as active (already visited) and inserted in the growing path at the current position. If the input vertex is the destination vertex the procedure records the path, labels the vertex as inactive and exits. Otherwise, the routine is recursively called on each vertex reached by any outgoing arc from the input vertex provided that the new vertex is not labeled as active (which would imply a cycle) and the length of the growing path is within the fixed threshold. The vertex count and the total length are increased and decreased before and after recursion.

The algorithm has time complexity $\mathcal{O}(q^{|V|+1})$, where q is an upper bound on the number of outgoing arcs from any vertex. The exponential complexity is due to the problem nature as explained in the following example. Consider the network, consisting of $n = 3h+1$ vertices and with $q = 3$, shown in Fig. 2.1. Consider vertex 1 as O_c and vertex $h+1$ as D_c and let $\epsilon \leq \tau$. The h arcs connecting the $h+1$ central vertices with length 1 define the shortest path from O_c to D_c , and, thus, $SP_c = h$. The remaining arcs have length $\frac{1}{2} + \frac{\epsilon}{2h}$. All possible paths in this network from O_c toD_c belong to the set K_c^τ. The number of the possible paths is $3^h = 3^{\frac{n-1}{3}}$.

Algorithm 1: Constructing Eligible Paths

Input : V, A, C, τ
$s \leftarrow ComputeShortestPathMatrix(V, A)$
for $i \in V$ **do**
  $status(i) \leftarrow INACTIVE$
end
for $c \in C$ **do**
  $Path \leftarrow \emptyset$
  $cPath \leftarrow 0$
  $lPath \leftarrow 0$
  $ScanVertex(O_c)$
end

Algorithm 2: ScanVertex

Input : i

Global variables : $s, \tau, status, Path, cPath, lPath$

```
status(i) ← ACTIVE
path(cPath) ← i
if i = D_c then
    Record(Path)
else
    for j ∈ δ+ do
        // j is the endpoint of an arc exiting from i
        if status(j) = INACTIVE and lPath + l_ij + s(j, D_c) ≤ (1 + τ)SP_c then
            cPath ← cPath + 1
            lPath ← lPath + l_ij
            ScanVertex(j)
            lPath ← lPath − l_ij
            cPath ← cPath − 1
        end
    end
end
status(i) ← INACTIVE
```

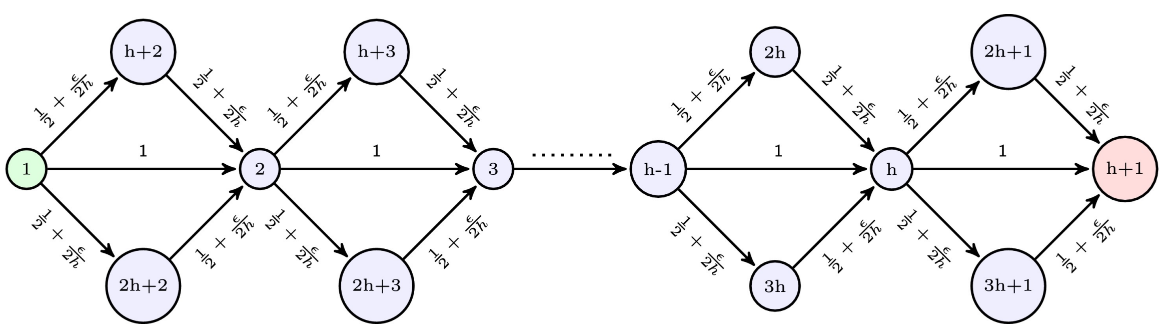


Figure 2.1: An instance demonstrating the exponential complexity of the algorithm for generating the sets of eligible paths for the OD pairs

2.2.3 Lower bounds on the minimum maximum arc utilization

In this section we propose an arc-based formulation for the congestion model where, for each OD pair c, all possible paths from O_c to D_c are implicitly considered. The model, which we refer to as the unconstrained congestion model, is equivalent to the congestion model in which the maximum travel inconvenience is set to $+\infty$. Allowing all paths for all the OD pairs means that the result of the unconstrained congestion model is a lower bound for the congestion model for any value of τ.

The unconstrained congestion model

$$\rho^*_\infty \equiv min \qquad \rho$$

subject to

$$\sum_{j \in V} f^c_{ij} - \sum_{j \in V} f^c_{ji} = 0 \qquad \forall c \in C, \qquad i \in V, \qquad i \neq O_c, \qquad i \neq D_c \tag{2.9}$$

$$\sum_{j \in V} f^c_{ij} - \sum_{j \in V} f^c_{ji} = d_c \qquad \forall c \in C, \qquad i = O_c \tag{2.10}$$

$$\sum_{j \in V} f^c_{ij} - \sum_{j \in V} f^c_{ji} = -d_c \qquad \forall c \in C, \qquad i = D_c \tag{2.11}$$

$$\rho \geq \frac{\sum_{c \in C} f^c_{ij}}{u_{ij}} \qquad \forall c \in C, \qquad \forall (i,j) \in A \tag{2.12}$$

$$f^c_{ij} \geq 0 \qquad \forall c \in C, \qquad \forall (i,j) \in A \tag{2.13}$$

$$\rho \geq 0 \tag{2.14}$$

Constraints (2.9) – (2.11) guarantee the flow rate conservation. The objective function together with Constraints (2.12) set the minimum maximum arc utilization to the minimum of the maximum value, over all arcs, of the ratio between the total flow rate and the arc capacity. Finally, constraints (2.13) and (2.14) define the domains of the decision variables.

Remark 4. *The value ρ^*_∞ on the unconstrained congestion model is a lower bound for ρ^*_τ,* *for any value of the maximum travel inconvenience τ, i.e.,*

$$\rho_\tau^* \geq \rho_\infty^* \qquad \forall \tau \tag{2.15}$$

Thus, the value ρ_∞^* is the minimum level of congestion that can be achieved when vehicles can be sent along any possible path (regardless of the experienced travel inconvenience). This value may be greater than one due to the structure of the network and high levels of the traffic.

Similar to the unconstrained congestion model, [25] studied a general form of *bottleneck network flow problem (BNFP)* which is defined on a directed graph $G = (V, A)$ where each arc a $\in A$ has a capacity and a weight and each node $v \in V$ is either supply, demand or transshipment node, associated with an integer value. The objective is to minimize the maximum weight on an arc with positive flow. The authors showed that this problem can be solved (by binary search threshold algorithm) as an $\mathcal{O}(logn)$ sequence of maximum flow problems where n is the number of nodes in the network. The maximum flow problems are defined on auxiliary graphs created from G where, in addition to the set of nodes, a dummy source and a dummy sink node are added to G to connect all supply and demand nodes respectively, and a subset of arcs in G is chosen from the ascending arrangement of arc weights, i.e., if an arc in the ascending arrangement is chosen, then all arcs whose weight is lower than the chosen arc's weight is considered in the problem. Similar approach and results were presented by [26] who investigated a bottleneck objective of minimizing maximum latency of flow-carrying arcs in static network problems, where each arc has a latency function that describes the common delay experienced by the flow on that arc as a function of the flow rate. They defined the bottleneck problem as identifying a feasible *bottleneck flow* that minimizes the maximum latency among arcs with positive flow for a given instance with multiple commodities. They showed that a bottleneck flow can be obtained by solving $|A|$ (number of arcs in the network) convex programs if the arc latency functions are convex. In this case, subset of arcs in each sub-problem are chosen from the ascending arrangement of arc latencies, when there is no flow on them. The authors showed that the value of bottleneck flow is equal to minimum value of convex programs. Applications of minimizing maximum latency can be found in network routing and evacuation literature [27].

In the following, we derive lower bounds on ρ^*_∞ that can be computed from the network parameters. The results we derive recall results that are well known for the maximum flow problem (see [28]).

Definition 1. *A cut-set for a set of OD pairs $\hat{C} \subseteq C$ is a minimal arc set $A_{\hat{C}}$ such that in the graph $G' = (V, A \backslash A_{\hat{C}})$ no path connecting O_c to D_c exists for any OD pair in $\hat{C}$. The capacity of a cut-set A is $u_{\hat{C}} = \sum_{(ij) \in A_{\hat{C}}} u_{ij}$.*

Definition 2. *A minimum capacity cut-set for $\hat{C}$ is a cut-set $A^*_{\hat{C}}$ with minimum capacity. We denote the capacity of $A^*_{\hat{C}}$ by $u^*_{\hat{C}}$.*

Theorem 1. *Let $A^*_{\hat{C}}$ be a minimum cut-set for any set of OD pairs $\hat{C} \subseteq C$, and let $u^*_{\hat{C}}$ be its capacity. Then, $\rho^*_\infty \geq \max_{\hat{C} \subseteq C} \frac{\sum_{c \in \hat{C}} d_c}{u^*_{\hat{C}}}$.*

Proof. Let $\bar{G}$ be a network in which the capacities $\hat{u}_{ij}$ are the capacities u_{ij} of the original network G multiplied by the minimum maximum arc utilization ρ^*_∞, i.e. $\hat{u}_{ij} = \rho^*_\infty u_{ij}$. Let $\hat{C} \subseteq C$ be a set of OD pairs and $A^*_{\hat{C}}$ a minimum cut-set for $\hat{C}$. In $\bar{G}$, it is possible to route the demand of the OD pairs in $\hat{C}$ without violating the capacities $\hat{u}_{ij}$. Hence, $\sum_{c \in \hat{C}} d_c \leq \sum_{(ij) \in A^*_{\hat{C}}} \hat{u}_{ij} = \sum_{(ij) \in A^*_{\hat{C}}} \rho^*_\infty u_{ij} = \rho^*_\infty \sum_{(ij) \in A^*_{\hat{C}}} u_{ij} = \rho^*_\infty u^*_{\hat{C}}$. Hence, $\frac{\sum_{c \in \hat{C}} d_c}{u^*_{\hat{C}}} \leq \rho^*_\infty$. As this inequality holds for any subset $\hat{C} \in C$, the claim follows. □

In the following, we show that in the case of a single OD pair equality holds, but that examples can be found with strict inequality for the case of multiple OD pairs.

Theorem 2. *Let $|C| = 1$ and let us denote by c the OD pair. Let A^*_C be a minimum cut-set for c with capacity u^*_c. Then, $\rho^*_\infty = \frac{d_c}{u^*_c}$.*

Proof. Let $\hat{\rho} = \frac{d_c}{u_c^*}$. We prove that $\rho_\infty^* = \hat{\rho}$ in two steps, showing that:

1. There exist a flow $\hat{f}_{ij}$. $\forall (i,j) \in A$,such that:
   - d_c flows from O_c to D_c,
   - $\hat{f}_{ij} \leq \hat{\rho}, \forall (i,j) \in A$,
   - $\hat{f}_{ij} = \hat{\rho}$ for some $(i,j) \in A$.

2. There does not exist a flow $g_{ij}, \forall (i,j) \in A$, such that:
   - d_c flows from O_c to D_c,
   - $\frac{g_{ij}}{u_{ij}} < \hat{\rho}, \forall (i,j) \in A$.

   (a) We solve the maximum flow problem on the graph G for the OD pair c. Let $\delta^+(O_c)$ be the set of the successors of O_c. From the max flow - min cut theorem, the maximum flow F_{max} is equal to the minimum cut-set capacity u_c^*. Flows f_{ij} in the optimal solution of the maximum flow problem are such that:

   - $f_{ij} \leq u_{ij}, \forall (i,j) \in A$,
   - $f_{ij} = u_{ij}, \forall (i,j)$ such that $(i,j) \in A_c^*$.

   We construct new flows $\hat{f}_{ij} = f_{ij}\hat{\rho}, \forall (i,j) \in A$, and obtain:

   - $\sum\limits_{j \in \delta^+(O_c)} \hat{f}_{O_c j} = \sum\limits_{j \in \delta^+(O_c)} f_{O_c j}\hat{\rho} = \hat{\rho} \sum\limits_{j \in \delta^+(O_c)} f_{O_c j} = \hat{\rho} F_{max} = \hat{\rho} u_c^* = d_c$. Thus this flow satisfies all the demand,
   - $\frac{\hat{f}_{ij}}{u_{ij}} = \frac{f_{ij}\hat{\rho}}{u_{ij}} \leq \hat{\rho}, \forall (i,j) \in A$,
   - $\frac{\hat{f}_{ij}}{u_{ij}} = \frac{f_{ij}\hat{\rho}}{u_{ij}} = \hat{\rho}, \forall (i,j)$ such that $(i,j) \in A_c^*$ where $f_{ij} = u_{ij}$.

   (b) By contradiction. Suppose there exists a flow g_{ij} such that $\frac{g_{ij}}{u_{ij}} < \hat{\rho}, \forall (i,j) \in A$. Then, $\sum\limits_{(i,j) \in A_c^*} g_{ij} < \sum\limits_{(i,j) \in A_c^*} u_{ij}\hat{\rho} = \hat{\rho} \sum\limits_{(i,j) \in A_c^*} u_{ij} = \hat{\rho} u_c^* = d_c$. As this flow is not able to satisfy all the demand, we have a contradiction.

$\square$

Remark 5. $\rho^*_\infty = \max\limits_{\hat{C} \subseteq C} \frac{\sum\limits_{c \in \hat{C}} d_c}{u^*_{\hat{C}}}$ *does not hold when there are two or more OD pairs in the network, that is when* $|C| \geq 2$.

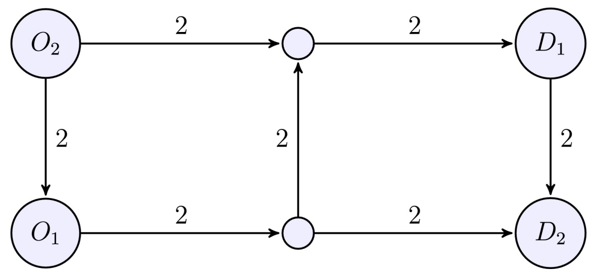


Figure 2.2: Example for two OD pairs

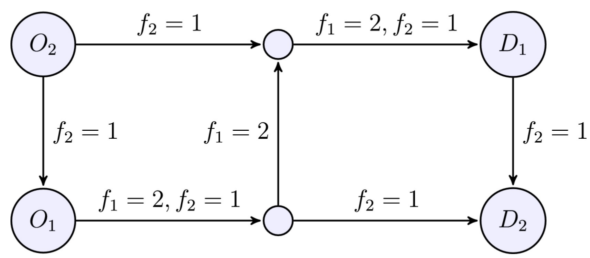


Figure 2.3: Optimal flows on counterexample graph

The example in Figure 2.2 where all arcs have capacity u equal to 2 and each OD pair had demand equal to 2 shows that the equality does not hold in the case of two OD pairs. Deriving a lower bound on the minimum maximum arc utilization means exploring all the possible subsets of OD pairs in C. The number of possible subsets in C is given by the cardinality of the power set of $C, P(C)$. The cardinality of the power set depends exponentially on the number of OD pairs contained in C, i.e. $|P(C)| = 2^n$, where n is the number of OD pairs in C.

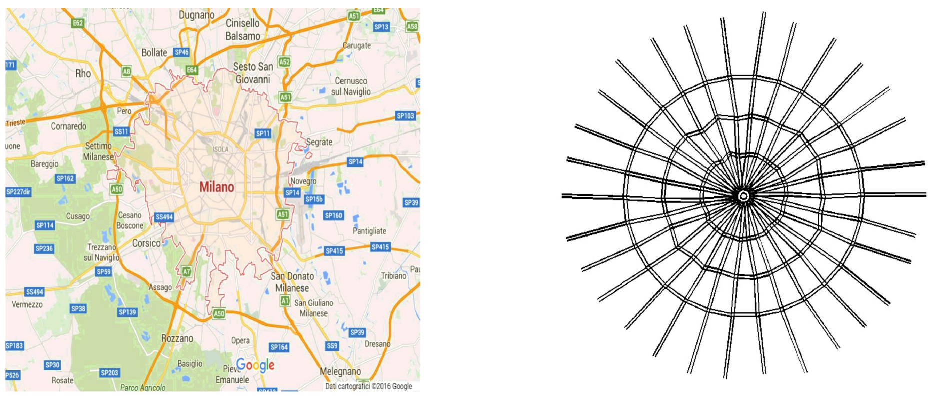
In Figure 2.3, the optimal flows for each OD pair are shown. It is easy to see that the demand is completely routed. The minimum maximum arc utilization value is equal to $\rho^*_\infty = \frac{3}{2}$. Considering each subset $\hat{C} \subseteq C$ and calculating $\rho^{\hat{C}}_\infty = \frac{\sum\limits_{c \in \hat{C}} d_c}{u_{\hat{C}*}}$ for each subset, we obtain that the maximum of all these values is $\max\limits_{\hat{C} \subseteq C} \rho^{\hat{C}}_\infty = \max\limits_{\hat{C} \subseteq C} \frac{\sum\limits_{c \in \hat{C}} d_c}{u_{\hat{C}*}} = 1 < \rho^*_\infty$.

2.3 Computational Results

The instance generator was implemented in Java, and the optimization models were solved by CPLEX 12.6.0. The generation of the instances and the experiments were run on a Windows 64-bit computer with Intel Xeon processor E5-1650, 3.50 GHz, and 16 GB Ram. Instance generation depends on a few controls, which are described in more detail in Section 2.3.1. Considering all possible combinations of these controls, the number of different types of networks is 16. Instance generation also depends on some random factors which are not included into the control set. For example, the coordinates of the nodes in a network is randomly perturbed from their original position. For each set of values for the controls, we generate 5 random instances. The total number of generated instances is therefore 80. The instances are available on http://or-brescia.unibs.it/instances. The statistics collected for each instance are described in Section 2.3.2. Detailed results and insights for a single, specific instance are presented and discussed in Section 2.3.3. Section2.3.4 is devoted to summary results for all instances.

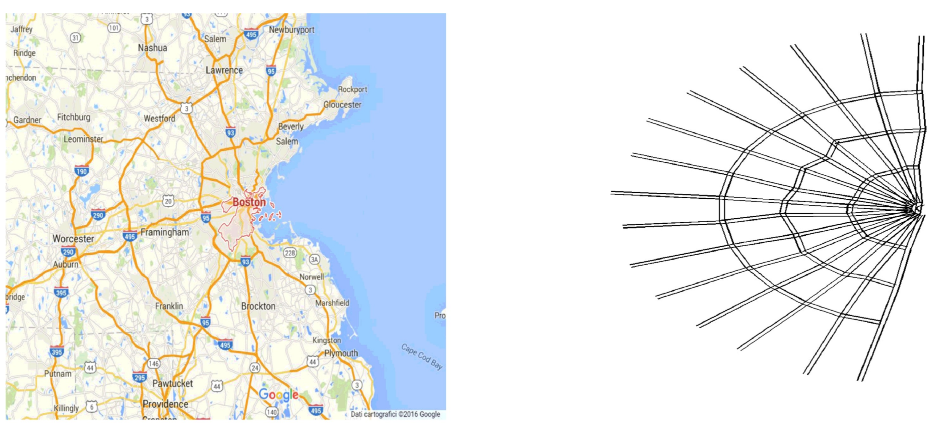
2.3.1 Instance Generation

We designed a procedure that generates instances reflecting the critical aspects of real city road networks. The procedure allows modeling of highways, bypasses, orbital roads and suburb streets, with capacities that depend on the road importance. The procedure can generate road networks with different shapes, for example with a circular structure (see Figure 2.4) or a semi-circular structure to be able to capture cities built adjacent to natural structures such as the sea or hills (see Figure 2.5). However, for the computational study, we restrict ourselves to instances with a circular structure as it represents the more general case. A circular road network has an inner ring, an outer ring, and some intermediate rings. The inner ring surrounds the central area and is considered a low-speed ring which allows access to the downtown area. The outer ring is a heavy traffic ring where all traffic coming from suburbs and surrounding towns converges. Depending on the city size, there can be one or more intermediate high-speed rings. Radial roads emanate outwards from the city center. Such roads finish at locations outside the outer ring. These locations are used in our experiments to represent origins (destinations) of traffic from (to) hinterland towns or highway connection points, i.e., the main roads entering the metropolitan area. The main roads entering the city are randomly classified as highways or high capacity roads. The radial roads also link the different rings.



(a) Map extract (b) City network

Figure 2.4: City with a circular road infrastructure



(a) Map extract (b) City network

Figure 2.5: City with a semi-circular road infrastructure

The instance generator has a number of parameters which allow us to generate different instances. In the following, we present groups of parameters that in combination define a major characteristic of the generated instance; we call such a group a *control*. We also name two specific sets of values of the parameters of a control to clearly identify different instance classes.

- **Control A : size** This control determines the size of the city. We distinguish small-to-medium cities and Big metropolitan areas:
  - Small-to-medium
    - Internal radius : 2000 m
    - External radius : 6000 m
    - Number of rings : 2
    - Number of radial roads : 15
    - Number of entering main roads : 2
  - Big
    - Internal radius : 1000 m
    - External radius : 20,000 m
    - Number of rings : 4
    - Number of radial roads : 30
    - Number of entering main roads : 6
- **Control B: angular width** This control determines the city shape. The generator takes as input a city angular width β measured in degrees. We generate only circular cities:
  - circular: $\beta = 360$
- **Control C: attraction points** In most cities, there are one or more attraction points, such as malls, stadiums, office parks, and schools. The generator takes as input the percentage ϕ of nodes that represent attraction points. We distinguish oligo-centric and poli-centric cities:

– oligo-centric: $\phi = 40\%$

– poli-centric: $\phi = 80\%$

- **Control D: traffic density** The flow for an OD pair is set by multiplying the sum over all origin outgoing arc capacities by a value σ, drawn uniform randomly from some specified interval. We distinguish off-peak and in-peak traffic densities for each network size:
  - Small-to-medium
    * off-peak: $\sigma \in [0.2, 0.4]$
    * in-peak: $\sigma \in [0.2, 0.6]$
  - Big
    * off-peak: $\sigma \in [0.1, 0.3]$
    * in-peak: $\sigma \in [0.3, 0.6]$
- **Control E: traffic density** Most of city traffic originates in the surrounding areas, but a percentage comes from locations inside the external ring. For this reason, the generator creates a percentage, γ, of OD pairs in C with both origin and destination inside the city. We distinguish low and high in-city traffic:
  - low : $\gamma = 10\%$
  - high : $\gamma = 20\%$

The capacity of each road is computed taking into account the following parameters:

- Number of lanes,
- Average speed,
- Safe separation distance.

We generate instance with four different classes of roads: Highways, main roads, secondary roads and links between roads. Each class has different values of the parameters listed above. In particular, highways have the highest number of lanes and the largest safe separation distance. while secondary roads have the smallest number of lanes and the smallest safe separation distance. Also the average speed value differs from class to class. We note that the average speed takes into account traffic lights and other traffic regulations. For the sake of simplicity, we have assumed that the speed is the same on all roads. In this case using distance of travel time as length is equivalent.

2.3.2 Statistics

In our experiments, we vary the value of the maximum travel inconvenience from $\tau = 0\%$ (all vehicles follow the shortest origin-destination path) to $\tau = 35\%$ (and the values of the compliance rate from 100% to 60%). We collect and compute the statistics shown in Table 2.2. To present large amounts of information in an easy-to-interpret and concise form, we mostly rely on graphs.

Table 2.2: Statistics

Network congestion	
ρ^*_τ	The minimum maximum arc utilization ρ^*_τ obtained from the congestion model.
$\frac{x_{ij}}{u_{ij}}$ classes	The fraction of arcs falling in each of the following four classes for different τ values:
	• unused arcs $\frac{x_{ij}}{u_{ij}} = 0$
	• unused arcs $0 < \frac{x_{ij}}{u_{ij}} \leq 1$
	• lightly congested arcs $1 < \frac{x_{ij}}{u_{ij}} \leq 1.5$
	• heavily congested arcs $1.5 < \frac{x_{ij}}{u_{ij}}$
No congestion τ	The no-congestion travel inconvenience, i.e. the minimum value of maximum travel inconvenience needed to avoid congestion.
User Experience	
Experienced Inconvenience	The weighted average experienced travel inconvenience,i.e.the optimal value of the inconvenience model.
Inconvenience classes	The percentage of the demand experiencing different levels of travel inconvenience:
	• level A: from 0% to 5%
	• level B: from 5% to 10%
	• level C: from 10% to 15%
	• level D: from 15% to 20%
	• level E: from 20% to 25%
	• level F: from 25% to 30%
	• level G: from 30% to 35%
OD paths	
Generated paths	Average number of generated paths per OD pair.
Selected paths	Selected paths in the optimal solution of the inconvenience model:
	• average number of selected paths per OD pair
	• maximum number of selected paths per OD pair

2.3.3 Detailed results and insights for a specific instance

In this section, we analyze the traffic patterns produced by the proactive route guidance approach for a single instance. We chose an instance with a large, circular, and oligo-centric city network with high in-city traffic and in-peak traffic density. In order to provide a detailed understanding of the potential benefits of adopting proactive route guidance, we solved the model with τ ranging from 0% to 35% in increments of 1%.

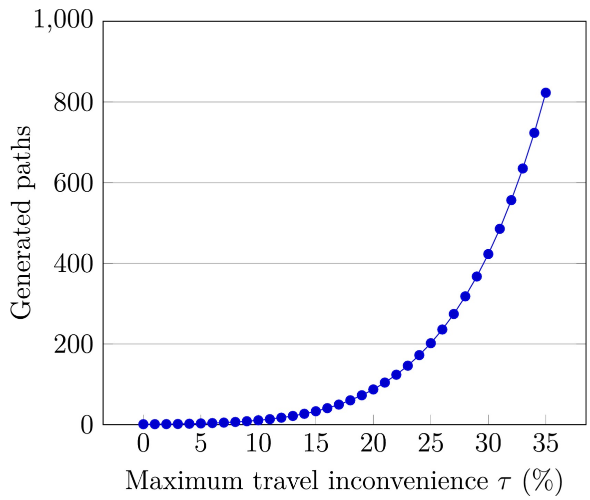


Figure 2.6: Average number of generated paths

In Figure 2.6, we show the average number of generated paths per OD pair. As expected, the average number of generated paths grows exponentially. Clearly, at higher values of the maximum travel inconvenience, there are more paths to choose from and it should be possible to reduce congestion more.

In Figure 2.7, we show the minimum maximum arc utilization ρ^*_τ for the different values of the maximum travel inconvenience τ. As expected, ρ^*_τ is a non-increasing function of τ. We note that for $\tau = 26\%$ the value of ρ^*_τ drops below 1, which implies that there is no congestion in the system. Consequently, for values of the maximum travel inconvenience greater than or equal to 26%, the proactive route guidance approach minimizes the weighted average experienced travel inconvenience while ensuring that the minimum maximum arc utilization does not exceed one. (We note that the experienced travel inconvenience value is realistic only when a path has no arcs with an arc utilization that exceeds one. Otherwise, the experienced travel inconvenience value represents an underestimate, as congestion will be encountered along the path.)

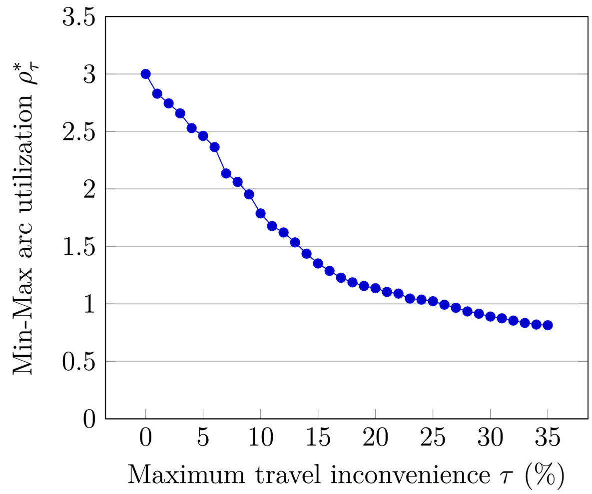


Figure 2.7: Minimum maximum (Min-Max) arc utilization

In Figure 2.8, we show the arc utilization distribution for the different values of the maximum travel inconvenience. We see that the fraction of unused arcs in the road network remains almost steady. When τ increases from 0% to 13%, the fraction of heavily congested arcs decreases, but, at the same time, the fraction of lightly congested arcs increases and the fraction of the non-congested arcs slightly decreases. We have no heavily congested arc for $\tau \geq 14\%$, and, as previously observed, the road network becomes free of congestion when $\tau \geq 26\%$. The two seemingly abrupt transitions from $\tau = 13\%$ to $\tau = 14\%$ when the road network looses heavily congested arcs, and from $\tau = 25\%$ to $\tau = 26\%$ when the road network looses all congested arcs, are smoother than they may appear. Indeed, for $\tau = 13\%$ we have $\rho^*_\tau = 1.54$ and all the 8.3% heavily congested arcs are very close to be classified as lightly congested. For $\tau^* = 25\%$ we have $\rho^*_\tau = 1.02$ and all the congested arcs are very close to be classified as non-congested.

In Figure 2.9 we show the weighted average experienced travel inconvenience for maximum travel inconvenience values greater than or equal to 26%, i.e., values for which

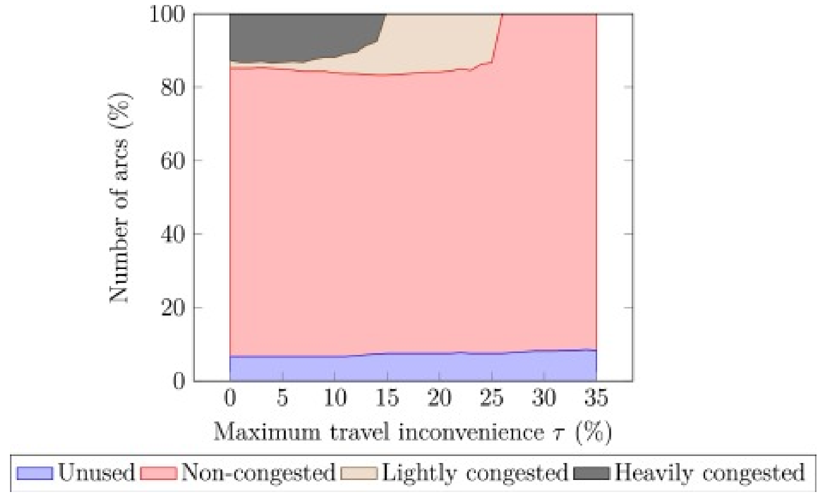


Figure 2.8: Arc utilization distribution

congestion can be eliminated. As expected, we see that the weighted average experienced travel inconvenience decreases, because additional paths are exploited to reduce the weighted average experienced travel inconvenience rather than to reduce the maximum arc utilization.

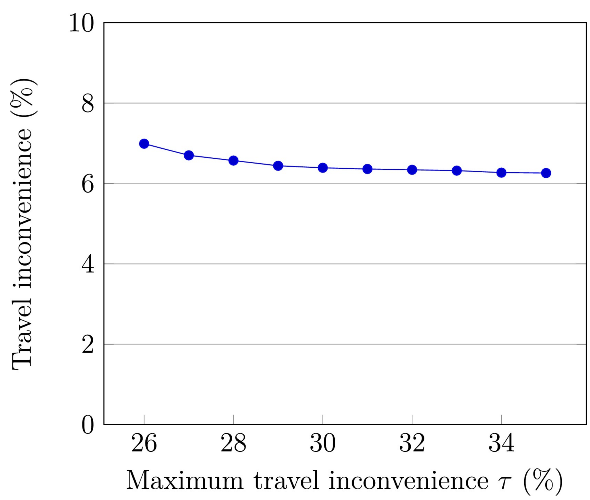


Figure 2.9: Weighted average experienced travel inconvenience

In Figure 2.10, we show, for $\tau = 26\%$, the number of OD pairs experiencing a certain level of travel inconvenience. Importantly, we see that for almost half of the OD pairs travel still occurs along the shortest path and that only for a relatively small number of OD pairs travel inconvenience exceeds 15%.

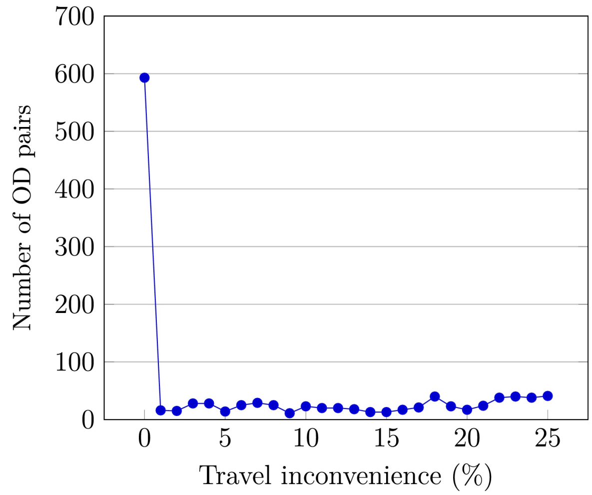


Figure 2.10: Experienced travel inconvenience at $\tau = 26\%$

The pattern shown in Figure 2.10 is also seen in Figure 2.11, where we show the fraction of OD pairs that is experiencing a certain travel inconvenience level (A through G) for values of $\tau \geq 26\%$. We see that most vehicles are sent along paths with little or no travel inconvenience (level A) and that only a small number of vehicles are sent along paths with a high travel inconvenience (levels E through G).

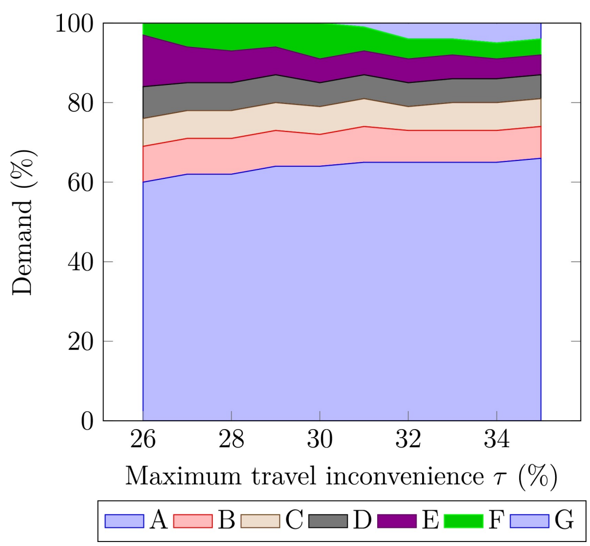


Figure 2.11: Experienced travel inconvenience distribution for various levels of maximum travel inconvenience

In Figure 2.12, we show the average and the maximum number of selected paths for an OD pair. We observe that the maximum number of paths selected for an OD pair is surprisingly small. For all but one of the maximum travel inconvenience values, the number of selected paths is less than or equal to three. This implies that vehicles with the same origin and destination are assigned to only a small number of different paths. (This will facilitate developing schemes that assign paths in a fair way over time, e.g., a round-robin assignment scheme.) We also observe that for all maximum travel inconvenience values, for most of the OD pairs, the vehicles are routed along a single path, as the average number of paths selected for an OD pair is very close to 1. However, we note that when a single path is selected for an OD pair, this path does not necessarily have to be the shortest path and that, in fact, in many cases it is not.

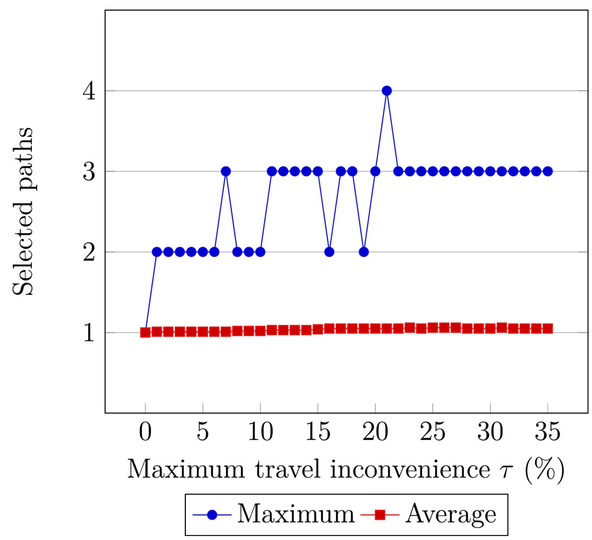


Figure 2.12: Selected paths

Even though we are focusing on an environment in which a specific origin-destination path can be assigned to each vehicle and in which that vehicle will follow the assigned path, it is informative to examine what happens when the 100% compliance assumption does not hold. Figure 2.13 shows the minimum maximum arc utilization ρ^*_τ that can be achieved as a function of maximum travel inconvenience for different compliance rates. We see that for $\alpha = 90\%$ congestion can still be avoided as the minimum maximum arc utilization falls below 1 at $\tau = 30\%$, but that for lower levels of compliance, this is no longer possible.

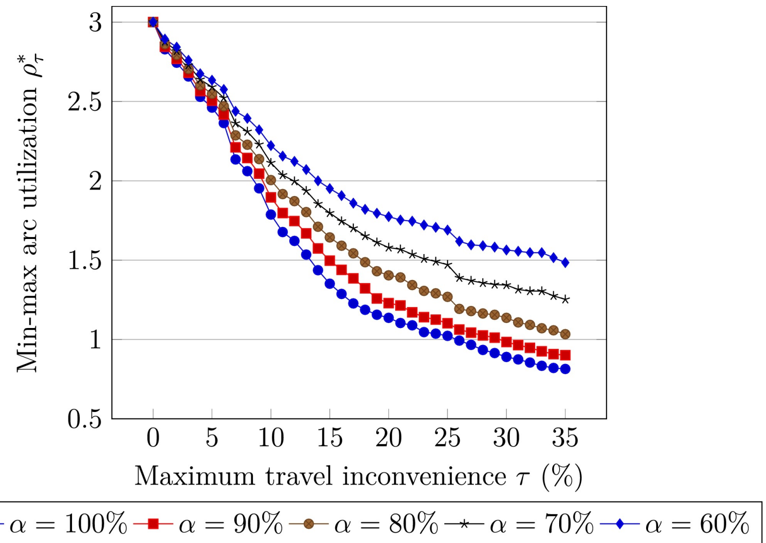


Figure 2.13: Impact of the compliance rate on the minimum maximum (min-max) arc utilization

2.3.4 Summary results for all instances

In this section, we present summary statistics for groups of instances. In Figures 2.14-2.16, we group instances that require the same maximum travel inconvenience to be able to reach the no-congestion state. In the final two charts, i.e., Figures 2.17 - 2.18, we look at all instances, but for specific maximum travel inconvenience values.

In Figure 2.14, we show for each value of the maximum travel inconvenience the number of instances for which it is possible to reach a no-congestion state (distinguishing in-peak and off-peak traffic densities). We observe that with off-peak traffic density at $\tau = 22\%$ it is possible to reach a no-congestion state for all 40 instances, but that with in-peak traffic density even at $\tau = 35\%$ it is not possible to reach the no-congestion state for two instances. However, the value of minimum maximum arc utilization for these two instance is very close to one, and a no-congestion state is attained at $\tau = 36\%$.

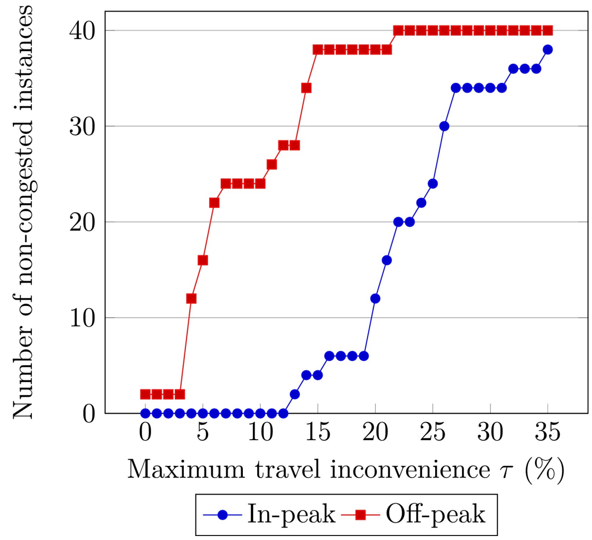


Figure 2.14: Number of instances in a no-congestion state for different values of $\tau(\%)$

In Figures 2.15 and 2.16, we consider particular statistics and average this statistic over all instances that reach the no-congestion state at a particular maximum travel inconvenience value. In Figure 2.15, we show the average and the maximum number of selected paths. We observe that the results suggest that when a higher maximum travel inconvenience is required to reach the no-congestion state, more OD pairs will be assigned more than one path as both the average of the maximum number of selected paths and the average of the average number of selected paths are (slightly) higher. In Figure 2.16 , we show the weighted average experienced travel inconvenience. We observe that, as expected, the weighted average experienced travel inconvenience is higher when it is more difficult to reach a no-congestion state, but also that the weighted average experienced travel inconvenience remains relatively small even for instances where it is challenging to reach a no-congestion state.

In Figure 2.17, we show the average minimum maximum arc utilization over all instances with in-peak and off-peak traffic densities. As expected, we see that the minimum maximum arc utilization is (significantly) smaller when the traffic density is smaller, as fewer vehicles have to be accommodated in the same road network. We see that with in-

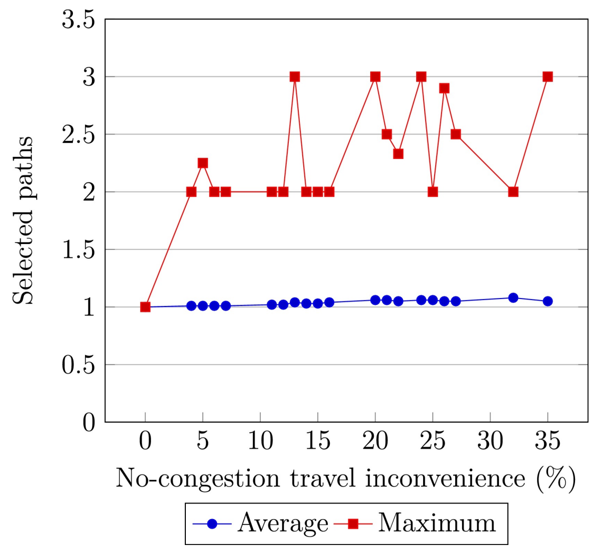


Figure 2.15: Number of paths chosen at the no-congestion travel inconvenience $\tau(\%)$

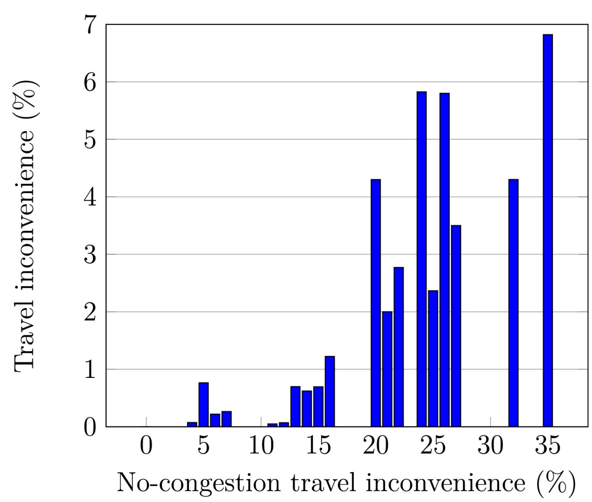


Figure 2.16: Weighted average experienced travel inconvenience at the no-congestion travel inconvenience value $(\%)$

peak traffic density it is possible (on average) to reach a no-congestion state at $\tau = 23\%$. On the other hand, with off-peak traffic density, a no-congestion state is reached (on average) at $\tau = 8\%$.

In Figure 2.18, we show the average arc utilization distribution. The average number of unused arcs remains almost the same while the average number of non-congested arcs steadily increases with increasing values of maximum travel inconvenience. At $\tau = 35\%$,

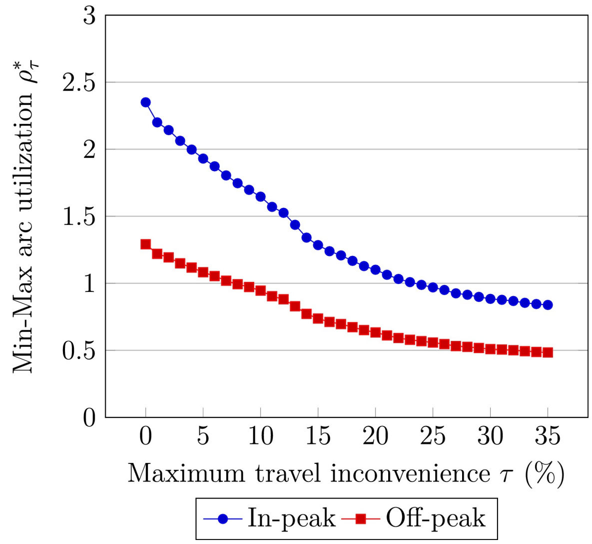


Figure 2.17: Minimum maximum (Min-Max) arc utilization

the proactive route guidance approach is able to eliminate heavily congested arcs in all instances and reduce the average number of lightly congested arcs to less than 0.5%. This congestion "residue" is due to two instances that remain lightly congested at $\tau = 35\%$.

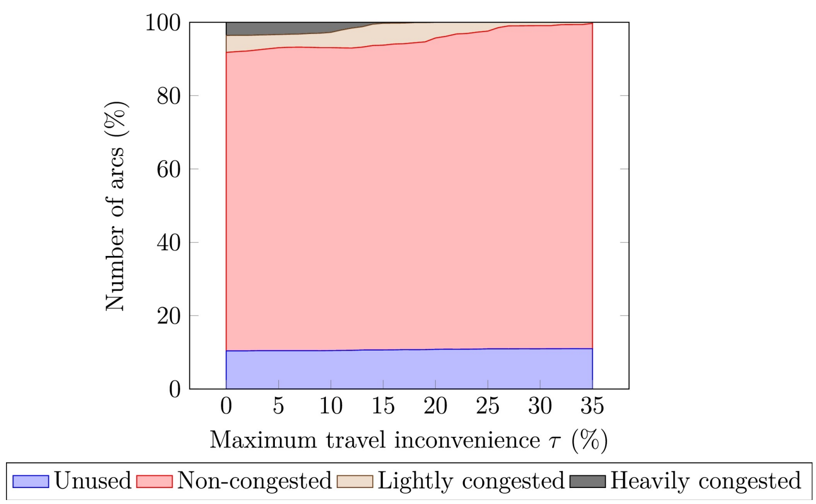


Figure 2.18: Arc utilization distribution

2.4 Conclusion

In this study, we have conducted a computational study of the potential of proactive route guidance to reduce or avoid congestion in an environment in which a specific origin-destination path can be assigned to each vehicle in the system and in which that vehicle will follow the assigned path. We believe that the advent of autonomous (or self-driving) vehicles makes this a (much more) realistic assumption.

Our proposed approach has an important (computational) advantage over previously proposed approaches: it relies only on linear programming. As such, it is more likely to scale and be usable in real-world settings. However, much needs to happen to make that a reality. One important step in that direction is the dynamic generation of paths (as opposed to generating the set of eligible paths upfront). Given the exponential growth of the sets of eligible paths when the maximum travel inconvenience increases, this will be critical.

Another critically important extension is incorporating traffic-dependent arc travel times, which can be done efficiently by employing piece-wise linear approximations of these functions. This will allow more accurate modeling of travel times along paths that contain one or more congested arcs, i.e., arcs with a utilization greater than one, which is especially important in situations where it is not possible to reach a no-congestion state.

A final extension, but one that is more involved, considers varying demand rates over time (instead of assuming steady- state behavior), which will allow more accurate modeling of the traffic dynamics.

CHAPTER 3
MARGINAL COST PATH - BASED GREEDY ALGORITHM FOR LARGE-SCALE SERVICE NETWORK DESIGN

3.1 Introduction

The long-haul transportation is the transportation operations that are mainly concerned with the movement of goods by rail, truck, ship or any combination of modes over relatively long distances between terminals or cities [29]. Throughout the history, the long-distance transportation has played a key role in the development and success of local, regional and global economies. It is the fast and cheap transportation of raw materials and finished products over long distances that contributed the economic growth during the Industrial Revolution and the success of the modern-day online retailers. It has connected businesses and individuals, creating new opportunities and channels for production, trade and consumption.

As it is not economically viable to transport each individual freight separately, the long-haul carriers such as railways, less-than-truckload (LTL) motor carriers, express package services and inter-modal container shipping lines rely on consolidation opportunities. To achieve the economies of scale, the freight is collected and consolidated from multiple shippers and routed through a network of consolidation terminals (e.g rail yards, ports, LTL break-bulks and end-of line). While the carriers can utilize their resources efficiently and reduce transportation costs with consolidation, they are faced with increased handling cost and time. In the pas, long delivery windows (e.g. week-long) have allowed carriers to identify and take advantage of profitable consolidation. With the introduction of lean production systems (e.g. just-in-time manufacturing) and competitive service level guarantees (e.g. same day delivery, next day delivery) offered by online retailers, carriers are challenged with the short service times that leave less time for consolidation. For instance,

Figure 3.1 shows the freight profile for a LTL carrier whose 80% of shipments are promised to be delivered within two days.

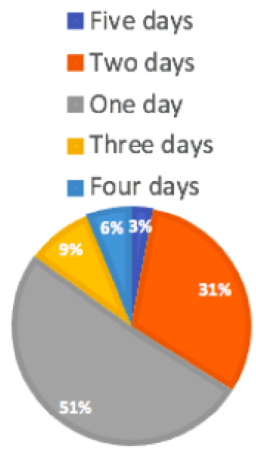


Figure 3.1: Freight profile for a LTL carrier

In this chapter, we focus on service network design problem to solve main tactical problems faced by small parcel express carriers which include cost-effective routing of less-than-truckload freight on a line-haul network. We generate operational freight schedules that take the following realistic complexities into account: i) Multiple tight service level guarantees, ii) Operational restrictions at line-haul terminals and iii) Multiple freight transfers from city operations to line-haul terminals throughout the day with varying volumes.

We develop a marginal cost path-based greedy heuristic that works with a partially time-expanded network to solve large scale real-life instances found in practice. Our approach involves two consolidation improvement heuristics and a iterative refinement heuristic algorithm inspired by [5], which iteratively eliminates infeasibilities introduced by the discretization of time and the representation of the timing of events. We conduct an extensive computational study for one of the largest Chinese small parcel carriers, SF Express.

The remainder of this chapter is structured as follows. Section 3.2 provides the detailed review of the literature on the service network design and relevant research. Section 3.3

describes how LTL system operates and Section 3.4 states the problem with the modeling choices and assumptions. Section 3.5 describes the solution approach in detail. Section 3.6 introduces the computational study. Finally, Section 3.7 discusses the contributions and future directions.

3.2 Literature Review

Decision making for a business entity is categorized into three levels: strategic, tactical and operational. Strategic decisions are usually long-term planning involving investment decisions (e.g. where to open a facility). Operational decisions are day-to-day decisions that involve high uncertainty. Tactical decisions serve as a bridge between the strategical and operational sides. Given the predetermined strategical decisions, tactical level focuses on short-term planning with interrelated decisions that are vital for the success of the day-to-day operations.

In the long-haul freight transportation, *Service Network Design* is used to designate the main tactical issues for the long-haul freight carriers: selection and scheduling of services, specification of terminal operations, routing of freight [29]. In literature, it has been widely used in a variety of freight transportation system such as trucking ([30], [31],[32], [33], [34], [35],[36],[37]), express shipment ([38], [39],[40],[41]), freight rail ([42],[43],[44],[45]) and multi-modal freight transportation ([46],[39],[47],[48]). It is often formulated as fixed charge multi-commodity network design problem for which it is not practical to solve real-life instances to optimality. Heuristic search techniques have been developed and combined with the exact optimization to construct and improve solutions for large-scale problem instances ([49], [50],[51],[52]). [53] gives an early survey on the network design problems and a detailed review on the problem formulations, modeling and solution approaches can be found both in [29] and [54].

In service network design, a *load plan* refers to the selection of paths for all anticipated freight on a terminal network to minimize the transportation costs incurred by moving the freight from its origin terminal to its destination terminal. Load plan design is especially critical for LTL carriers who rely on freight consolidation to increase their trailer utilization and cost savings. The first study on LTL load plan design was conducted by [31] followed by [30] [32] and [33]. In [31], Powell modeled the problem on a flat (static) network that did not consider service guarantees and timing of consolidation opportunities explicitly. Instead, he imposed a lower bound on the number of weekly trailers to be dispatched on any direct lane chosen in the load plan. This was sufficient to identify consolidation opportunities and meet a five-day service guarantee.

With the introduction of tighter service guarantees (e.g. same day, one-day service guarantees), static network models started to fall short of representing the service requirements and consolidation timing accurately. Hence, the focus in service network design shifted to dynamic networks that incorporate the timing of events such as departure and arrival of dispatches. Dynamic networks are the time-expanded version of static networks that are constructed with the copies of static network's nodes at discrete time points. In service network design, a node in a dynamic network usually represents a terminal at a point in time and an arc represents either the movement of freight from one terminal to another or the waiting of the freight at a terminal. Network design problems on dynamic networks can be formulated and solved using integer programming models. The size of the integer programs and solution quality depend directly on the time-discretization the schemes. Fine discretization of time represents the timing of activities accurately while leading computationally intractable integer programs. On the other hand, coarse discretization schemes result in smaller integer programs that can generally be solved to optimality, but often produce poor approximations to the continuous-time version of the problem in which time is modeled in such a way that accurately identifies consolidation opportunities (e.g. 1-minute time discretiztion).

[55] addressed the shortcomings of the static models and presented a dynamic model for LTL service network design with 15 terminals and 18 time points. [34] presented an

integer programming model on a dynamic network where a commodity path was represented in time and space. They represented time in daily granularity and considered a single time point for each day. [56] used a fine non-uniform time discretization to identify consolidation opportunities under tight service standards faced by LTL carriers. They used relatively detailed time-space modeling of dispatches between terminals compared to dispatches from break-bulk terminals which still allow them to capture consolidation opportunities in time accurately. Their research was the first study to consider day-differentiated load-plans which specified a unique load path for a commodity between an origin and destination pair originating on the same weekday.

While there were studies that focused on using finer time-discretization to improve the solution quality in service network design problems, a study by [5] investigated whether it is possible to solve the continuous-time version of the problem without modeling the problem on a fully time-expanded network. They developed a dynamic discretization algorithm which repeatedly solves service network design problem on partially time-expanded networks and manipulates the partially-time expanded network by analyzing the obtained solution. They showed that their iterative refinement algorithm produces the optimal continuous-time solution and they showed that it performs well in practice by conducting an extensive computational study.

The load plan design decisions give rise to challenging large-scale network optimization problems. As exact solution methods often fail to find the optimal solution, heuristic algorithms were proposed to find high-quality feasible solution. [33] modeled the problem on a flat (static) network and proposed a local improvement heuristic based on direct lane add-drops from the load plan. [34] presented an integer programming model and used a slope-scaling and load-planning tree generation techniques to produce high-quality solutions for large-scale instances. [56] solved a large scale load-planning problem using an iterative IP-based large-scale neighborhood search heuristic. At each iteration, a restricted load planning IP was solved to improve the load plan for freight destined for a single terminal. Their flexible load plans that captured predictable variations in origin-destination pairs by day of week lead to greater cost savings compared to the traditional load plans with the uniform in-tree structure. [36] further investigated this IP-based local search to obtain high-quality load plans. Rather than using in-tree neighborhood search, at each iteration, their heuristic algorithm re-optimizes the load plans for freight destined to multiple terminals. Compared to the in-tree neighborhood search suggested by [56], this search heuristic resulted in better cost savings and trailer utilization. As a common modeling choice, [36] and [56] considered freight to enter the line-haul network only at a single time point in a day.

3.3 LTL System Description

There are two types of inter-related operations for freight transportation in a LTL system: *city operations* and *line-haul operations*. City operations are concerned with pick-up and delivery of freight at customer locations in a small geographical region (e.g. city). Line-haul operations are mainly concerned with consolidation and routing of freight through a fixed line-haul network. A line-haul network has two types of terminals: end-of-line terminals and break-bulk terminals. End of line terminals serve only as origin and destination terminals. Freight collected by city operations enter to line-haul network at end-of-line terminals (serving as origin terminals) and routed line-haul freight transferred to city operations at end-of-line terminals (serving as destination terminals) for intra-city delivery. Break-bulk terminals are intermediate consolidation terminals in the line-haul network. Freight visiting a break-bulk terminal is unloaded from its inbound trailer and reloaded to outbound trailers along with other freight to be dispatch on the same direct lane, i.e. freight is *cross-docked*.

3.4 Problem Definition

We investigate the service network design problem faced by small parcel carriers with the following modeling complexities.

1. *Service classes*: Small parcel carriers offer various service classes to their customers. Each service class has a promised time of delivery which implicitly specifies a due time for each freight at its destination terminal in the line haul network such that intra-city operations are not delayed. These offerings may depend on the freight origin-destination pair, and therefore, may not be available to each freight.

2. *Freight arrival to the line-haul network*: With the introduction of tight service guarantees, small parcel carriers started to increase the frequency of their daily shipments from city operations to the line-haul network in order to efficiently use the sorting capacity at the line-haul terminals. As drivers in city operations pick-up shipments from customers during the day, they pass the collected freight (e.g. in hourly intervals) to the regional end-of-line terminal. Thus, depending on the collected volume, freight volume entering the line-haul network can show significant variability by the hour-of-day and day-of-week.

3. *Commodities*: When a freight arrives to its origin end-of-line terminal, it is sorted by the destination terminal before it is dispatched across the line-haul network. Once sorted by destination terminal, freight with common service level guarantee are further grouped together and considered a single *commodity*. Let K represent the set of commodities. Each commodity $k \in K$ is then characterized by $(o_k, d_k, e_k, l_k, q_k)$, where o_k is the origin terminal, d_k is the destination terminal, e_k is the earliest possible dispatch time at origin terminal, l_k is the latest possible arrival time (due time) at destination terminal and q_k is the weight associated with the commodity.

4. *Cross-docking at break-bulk terminals*: Upon visiting a break-bulk terminal, a com-

modity is first unloaded from the incoming vehicle, then consolidated with eligible commodities at the terminal and finally loaded into one of the outbound vehicles. This process is called cross-docking and δ_k represents the cross-docking time of commodity $k \in K$.

5. *Commodity waiting at a terminal* : Terminals have fixed capacity for handling (sorting / cross-docking) the incoming commodities and holding the already processed ones. To capture these capacity restrictions at terminals, including the origin terminals, each commodity is allowed to wait at any terminal for a maximum amount of time, τ_W, for further consolidation.

6. *Operating periods at a terminal*: Each terminal in line-haul network is active for a pre-defined set of operating periods during the day throughout the planning horizon. Associated with each operating period of a terminal is a *cut-off* time representing the latest arrival time for a freight to be processed (cross-docked) within that operating period. Any commodity that arrives to a terminal after this time point has to wait the next operating period to be processed.

7. *Planning horizon*: LTL freight volumes show significant variability between week days. On the other hand, they are typically similar across some period of time (e.g. weeks). Hence, a *n*-day planning horizon that is wrapped around the beginning of the chosen planning horizon to represent the repeating nature of the routing decisions is used to create day-differentiated load plans.

8. *Vehicle types*: Carriers utilize either company-owned or outsourced vehicles of different types with varying capacities. Depending on the lane, business type and operation schedule (e.g. one-way, out-and-back) these vehicles have different costs per unit distance.

Given these modeling complexities, we make the following assumptions for solving the service network design problem.

- There are as many vehicles as needed.
- The time that freight enters the system in the form of a commodity is taken to be the first time that it is available for dispatch, i.e. after origin sorting has taken place.
- In this chapter, only company-owned vehicles are considered and the vehicle costs per unit distance are given for one-way moves. We will cover outsourcing in Chapter 4.
- The volume of each commodity is integer.
- Every commodity can be dispatched in such a way that it meets the service promise (due time at the destination terminal), i.e. no commodity is allowed to arrive late.
- Holding and handling (e.g. sorting/cross-docking) commodities at a terminal do not incur any cost.
- A vehicle dispatch can arrive, wait and depart outside the operating periods of a terminal.
- The cross-docking time is same for each commodity, i.e. $\delta = \delta_k \; \forall k \in K$ and it is considered part of the commodity waiting time at a terminal.

The objective is to find a path, π_k, for each individual commodity $k \in K$ from its origin terminal to its destination terminal in the line-haul network while meeting service requirements, satisfying operational constraints and minimizing the transportation costs.

3.5 Solution Approach

Let $LN = (U, L)$ be the carrier's line-haul network with the node set U representing the set of of terminals in the network and the arc set, L, is the set of potential directs connecting terminals. For each terminal $u \in U$, a set of operating periods, B_u, is specified. An operating period for terminal $u \in U$, $b_u \in B_u$ is characterized by $(s_{b_u}, e_{b_u}, c_{b_u})$, where s_{b_u}

is the start time, e_{b_u} is the end time, c_{b_u} is the cut-off time associated with the operating period.

Let V represent the set of vehicle types available at carrier's disposal. The capacity of each vehicle type $v \in V$ is given by Q_v. There are one-way cost per unit distance, c_{lv} associated with each direct $l \in L$ and vehicle type $v \in V$. Let γ_l be the length and t_l be the travel time on direct $l \in L$.

A dispatch $\vartheta \in \Theta$ is characterized by $(o_\vartheta, dt_\vartheta, d_\vartheta, at_\vartheta, v_\vartheta)$ where o_ϑ is the origin terminal, dt_ϑ is the departure time at origin terminal, d_ϑ is destination terminal, at_ϑ is the arrival time at destination terminal, v_ϑ the vehicle type and K_ϑ is set of commodities carried on the dispatch ϑ.

We model the timing of activities such as vehicle dispatch times, commodity waiting and handling (sorting / consolidation) times at terminals using a time-expanded version of the line-haul network LN. Given a set of time points, T_u, considered for each terminal $u \in U$, the time-expanded line-haul network $TE\text{-}LN = (N, A)$ is composed of the set of timed-nodes, N and the set of timed-arcs, $A = A_D \cup A_H$, that connects the timed-nodes. Each node $n = (u, t) \in N$ represents a terminal $u \in U$ at a time point $t \in T_u$. Each arc $a = (n_1, n_2) = ((u_1, t_1), (u_2, t_2)) \in A_D$ represents a potential dispatch from terminal u_1 at time $t_1 \in T_{u_1}$ on direct (u_1, u_2) arriving at terminal u_2 at time $t_2 \in T_{u_2}$ when $u_1 \neq u_2$ and $a \in A_H$ represents holding a commodity at terminal u from time $t_1 \in T_u$ to t_2, the next point in T_u after t_1, when $u_1 = u_2 = u$. Figure 3.2 illustrates a time-expanded network of three terminals $(U1, U2, U3)$ with a set of timed-nodes specific for each terminal and timed-arcs representing commodity dispatch from terminal $U1$ to terminal $U2$ and commodity holding at terminal $U3$ for a wrapped planning horizon.

On the line-haul network, LN, a commodity $k \in K$ can either be routed on a direct $l = (o_k, d_k)$ or it can visit break-bulk terminals before reaching its destination terminal. Let F_k be the set of all possible flat paths for commodity $k \in K$ on LN. A flat path $f_k = (u_1, ..., u_{|f_k|}) \in F_k$ is a sequence of directs which starts at a commodity's origin terminal,

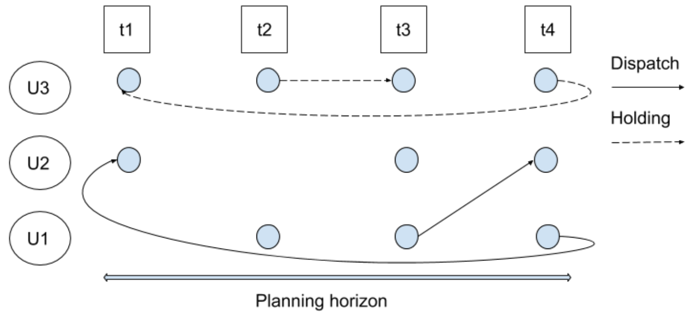
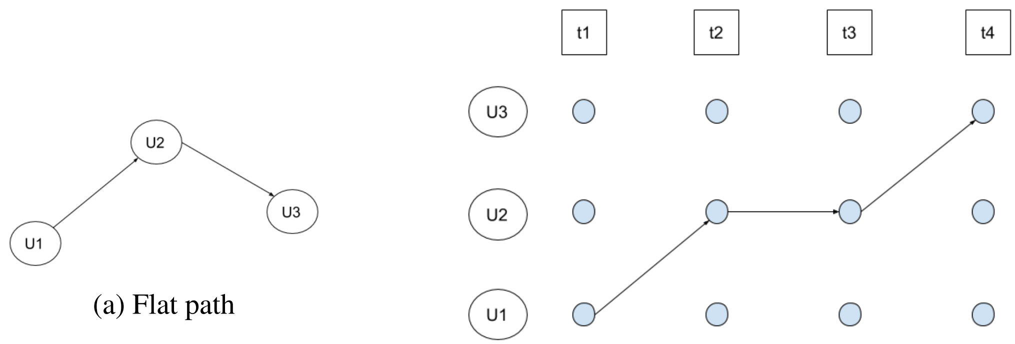


Figure 3.2: A time expanded line-haul network



(a) Flat path

(b) One possible of timed copy of flat path in (a) on a time-expended network

Figure 3.3: Flat versus timed-path

$u_1 = o_k$, and ends at its destination terminal $u_{|f|} = d_k$. When mapped onto $TE\text{-}LN$, each flat path f_k becomes either infeasible due to service and operational constraints or it has a corresponding set of time-feasible paths, P_{kf}. Hence the set of all time-feasible paths for commodity k is $P_k = \cup_{f_k \in F_k} P_{kf}$. Each time-feasible path $p_k = (a_1, ..., a_{|p_k|}) \in P_k$ is defined by a sequence of timed-arcs in A. Figure 3.3a shows a flat path and Figure 3.3b shows one possible timed copy of the same path on a time-extended network.

For any event e (e.g. arrival at a terminal $u \in U$) that takes place at time τ, if there is no time point with the same time in T_u, τ is mapped to an existing time point in T_u. We consider three mapping schemes: *optimistic*, *pessimistic* and *nearest*. These schemes are described in Table 3.1 and Figure 3.4 visualizes the mappings.

Independent of the choice of mapping scheme, for each commodity k, we relax the

Table 3.1: Mappings

Mapping	Description
Optimistic	$\max\limits_{t \in T_u}\{t \vert t \leq \tau\}$
Pessimistic	$\min\limits_{t \in T_u}\{t \vert t \geq \tau\}$
Nearest	$min(\vert\tau - \max\limits_{t \in T_u}\{t \vert t \leq \tau\}\vert, \vert\tau - \min\limits_{t \in T_u}\{t \vert t \geq \tau\}\vert)$

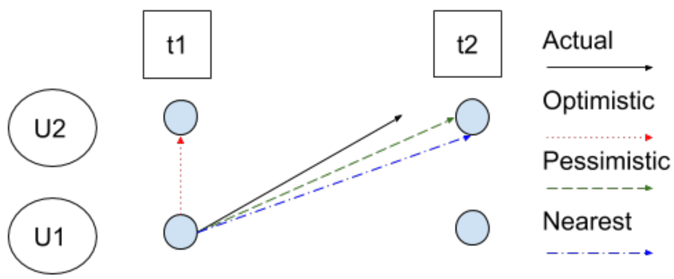


Figure 3.4: Mapping schemes

earliest possible dispatch time by setting it to the largest timed-node at o_k that is smaller than e_k.

$$e_k^* = \max_{t \in T_{o_k}}\{t \vert t \leq e_k\} \tag{3.1}$$

3.5.1 Marginal cost path-based greedy algorithm

It is possible to formulate an integer program that captures the problem-specific constraints. However, it will be challenging to solve the realistically sized instances. Thus, we choice to develop a marginal cost path-based greedy heuristic which determines a time-feasible path for each commodity $k \in K$ on a partially time-expanded line-haul network, $TE\text{-}LN$, specified by the line-haul terminals, U, the set of time points, T_u for each terminal $u \in U$ and a mapping scheme.Our objective is to minimize the total transportation cost, TC described as

$$TC = \sum_{a \in A_D} \sum_{v \in V} c_{lv} \gamma_a n_{av} \tag{3.2}$$

where n_{av} represents the number of planned vehicle dispatches of type $v \in V$ on arc a and γ_a is the length of the arc a.

We begin by constructing an initial load plan that specifies a time-feasible path for each commodity. First, we randomly choose a time-feasible path from a restricted set of time-feasible paths we calculate for a subset of randomly chosen commodities. For each commodity, the restricted set of time-feasible paths includes paths where no commodity waits at a terminal more than τ_W^-, where $\tau_W^- << \tau_W$. For the remaining commodities, we determine paths in a greedy and iterative fashion. In each iteration, we consider a single commodity and choose one of its least marginal cost time-feasible paths. Given the previously chosen commodity paths, the least marginal cost time-feasible path for a commodity is the path to which adding the commodity increases the total transportation cost the least.

Once we identify a time-feasible path for each commodity, we attempt to improve the load plan generated in the construction part. We make multiple improvement passes where we investigate the neighborhood of the current feasible solution to minimize the total transportation cost. In each pass, we remove one commodity at a time from its current path and assign it to its least marginal cost time-feasible path without modifying the paths for the rest of the commodities. We only change the time-feasible path for a commodity when the change improves the total transportation cost.

Except for the initialization step in the construction part of the algorithm, we process commodities in the increasing order of their *routing flexibility* on the line-haul network. Let l_k be the direct lane between o_k and d_k and let t_{l_k} be the travel time on it. Then, the routing flexibility of commodity k, ϕ_k, is defined by

$$\phi_k = (l_k - e_k) - t_{l_k} \tag{3.3}$$

As it is harder to find consolidation opportunities for commodities with low *routing flexibility* we choose paths for such commodities in early stages of the algorithm and align the paths for more flexible commodities with them to improve the transportation costs. In case of ties in *routing flexibility*, we process the commodities in decreasing order of the commodity volumes. It is harder to find available capacity and route commodities with high volume at no cost. We are more flexible with our path choices for commodities with low volume as they can more likely fit into the vehicles that we have already chosen for the previous commodities. The steps of the greedy heuristic is given by Algorithm 3.

Algorithm 3: Marginal cost path-based greedy algorithm for load planning

// Construction

Select a subset of commodities (K_i) randomly.

for $k \in K_i$ **do**

  Determine a subset of time-feasible paths restricted by the waiting time at terminals.

  Select one of the time-feasible paths from the restricted set randomly.

end

Sort the remaining commodities $(K_- = K \backslash K_i)$ in the increasing order of *routing flexibility*. In case of ties, sort the commodities in descending order of volume.

for $k \in K_-$ **do**

  Select the least-cost time-feasible path for commodity k.

end

// Improvement passes

for $m = 1, ..., M$ **do**

  Sort all commodities in the increasing order of *routing flexibility*. In case of ties, sort the commodities in descending order of volume.

  for $k \in K$ **do**

    Remove commodity k from its time-feasible path.

    Select the least-cost time-feasible path for commodity k.

  end

  // Consolidation improvement attempts

  Eliminate vehicles by vehicle utilization.

  Eliminate vehicles by dispatch alignment.

end

Marginal cost of a dispatch arc

Let ω_a be the total commodity volume and $C_a(\omega_a)$ be the total cost per unit distance on dispatch arc a before including commodity k. Then, the marginal cost of dispatch arc a for commodity k, λ_a^k, is the direct distance, γ_a, times the difference between the total cost per unit distance before and after adding commodity k to a, i.e.

$$\lambda_a^k = (C_a(\omega_a + q_k) - C_a(\omega_a))\gamma_a. \tag{3.4}$$

Assuming the total commodity volume on a dispatch arc can be split across vehicles, we find the optimal number of vehicles of each type to carry a total commodity volume, w, that minimizes the total cost per unit distance on dispatch arc a, $C_a(\omega)$, by the following dynamic program:

$$C_a(\omega) = \begin{cases} \min_{v \in V \wedge \omega \leq Q_v} \{c_{lv} + C_a(\omega - Q_v)\}, & \omega > 0 \\ 0, & \omega = 0 \end{cases}$$

The cost of routing a new commodity k on a time-feasible path $p_k \in P_k$, Λ_{p_k}, is the sum of marginal costs of dispatch arcs on the path p_k.

$$\Lambda_{p_k} = \sum_{a \in p_k} \lambda_a^k \tag{3.5}$$

The least marginal cost path for the commodity k, π_k^* is given by

$$\pi_k^* = \operatorname*{argmin}_{p_k \in P_k} \Lambda_{p_k} \tag{3.6}$$

Each time we search for the least marginal cost time-feasible path for a commodity, we need to calculate the marginal costs for dispatch arcs. In order to speed up the search, we memoize previously observed integer dispatch volumes and the corresponding vehicle combinations that will carry them with the minimum transportation cost on each direct lane $l \in L$.

Least marginal cost time-feasible path

Given the previously chosen set of paths for a set of commodities, we identify a least marginal cost (cheapest) paths to route the next commodity k from its origin terminal to its destination terminal while making feasible connections between terminals and meeting due time at its destination terminal. Depending on its *routing flexibility*, a commodity k can reach multiple timed-nodes of destination terminal, d_k, in $TE\text{-}LN$. Finding a cheapest path for commodity k is same as finding a shortest path with non-negative weights from a single origin to one of the nodes in a set of potential destinations while satisfying problem specific constraints.

We use an algorithm similar to Dijkstra's algorithm [57], where we account for constraints specific to our problem (e.g. cross-docking requirement at intermediate terminals, holding limit at terminals, having multiple timed-nodes of destination terminals). We maintain a priority queue of nodes using the data structure, binary (minimum) heap, and a set of visited nodes. An entry to the heap is characterized by $(n, C_n, R_n, H_n, Pred(n))$, where $n \in N$, C_n is the cumulative cost incurred after reaching node n, R_n is the remaining available time to reach the destination terminal from node n, H_n is the total time commodity held at the terminal $u \in U$ associated with n and finally $Pred(n)$ is the predecessor node of node n, used for determining the path at the end of algorithm. We initialize the heap by $(n = (o_k, e_k), 0, l_k - e_k, 0, -1)$.

Using the cost as the priority of a node, we successively extract the minimum cost node from min-heap. Each time we extract a node from heap, we add the node to the set of visited nodes if it is not already visited, otherwise we discard the node. If the extracted node is the destination terminal, the algorithm terminates with a cheapest path otherwise we examine all reachable nodes from it. If the service requirements and operational constraints (cross-docking requirement at break-bulk terminals, waiting time limit at terminals, due time at destination terminals) are not violated, we insert the reachable nodes to the heap. If heap becomes empty, it means there is no feasible path for the commodity.

Proposition 1. *If a node n is added to the list of visited nodes, C_n is equal to the cost of the cheapest path from source node (o_k, e_k) to node n for commodity k.*

Proof. Proof by contradiction. Assume that a node n is added to the list of visited nodes and C_n is not the cheapest path cost from source node to node n. Then, it means either (1) there should be another heap entry for node n with smaller cost or (2) such an entry has not been added to the heap yet.

(1) From the min-heap property, we know each entry has a cost greater than or equal to C_n when n was extracted from the heap.

(2) If there was another path to node n which leads to smaller cost than C_n, due to the exploration rule (nodes with smaller cost first), it should have been explored before the current entry node n and it should have been already in the list of visited nodes. □

Corollary 1. *It is sufficient to visit each node at most once.*

Proof. In each step, we expand the search from the cheapest reachable node. Hence, the cost of a node extracted from min-heap is always greater than or equal to the cost of the previous extracted nodes. If a node, say n, is in the list of visited nodes and becomes the cheapest reachable node at some step in the procedure, there is no need to process the node n since it will lead to including node n's reachable nodes to the heap with higher costs. □

3.5.2 Vehicle elimination by vehicle utilization

It is not always possible to eliminate low-utilized vehicle dispatches by sequentially removing commodities from their current paths and assigning them to their least-cost time-feasible paths. Hence, we also investigate eliminating an entire vehicle dispatch ϑ with low utilization by removing all commodities on ϑ from their current paths.

Let K_a represent the set of commodities carried on dispatch arc $a \in A_D^*$, UT_a represent the utilization of dispatch arc $a \in A_D^*$ ($UT_a = \frac{\sum_{k \in K_a} q_k}{\sum_{\vartheta \in \Theta_a} Q_v}$), Θ_a be the set of trucks scheduled on dispatch arc $a \in A_D^*$ and K_ϑ be the set of commodities carried on vehicle dispatch $\vartheta \in \Theta$. Among the set of dispatch arcs (A_R) in which each arc utilization is less than a chosen utilization threshold and the each arc distance is greater than a chosen distance threshold, we choose a dispatch arc randomly and eliminate the least utilized vehicle on it, if it improves the total cost.

Algorithm 4 describes the steps of the vehicle elimination procedure. $\Re$ represents the number of vehicle elimination trials.

Algorithm 4: Vehicle elimination by vehicle utilization

for $r = 1, ..., \Re$ **do**
- Randomly pick a dispatch arc $a \in A_R$.
- Find the vehicle dispatch $\vartheta \in \Theta_a$ that is that is not fully loaded.
- Remove vehicle dispatch ϑ by removing all commodities carried on ϑ from their current paths.
- Sort all the removed commodities (K_ϑ) in the increasing order of *routing flexibility*. In case of ties, sort the commodities in descending order of volume.
- **for** $k \in K_\vartheta$ **do**
  - Select the *least-cost time-feasible path*.
- **end**
- Keep the solution if total cost improves, otherwise revert the changes.

end

3.5.3 Vehicle elimination by dispatch alignment

In given a load plan, commodities with high *routing flexibility* might be routed on paths that arrive at their destination terminal earlier than their due times. This creates the opportunity to change the actual departure times of the dispatches that these commodities belong to. We may then align and consolidate commodities scheduled on each direct lane in time and decrease the total transportation cost by eliminating the low-utilized vehicles.

To align commodities in time, we first need to determine how much we can move the departure time of a vehicle dispatch $\vartheta \in \Theta$ such that commodities on ϑ can still make feasible connections between terminals, satisfy holding time limit at terminals and meet their due times at their destination terminals. [35] introduced a linear programming model that identifies a dispatch window for each dispatch constructed. Their goal was to identify dispatch windows that provided the most flexibility. Therefore, they maximized the sum of the widths of the individual windows for each dispatch in their linear program. We modify this model to take (i) waiting time limit at terminals, (ii) cross-docking time at break-bulk terminals and (iii) wrapped planning horizon into account. Table 3.2 summarizes the notation used in our dispatch window selection model.

Table 3.2: Notation for dispatch window selection model

Sets	
K	Set of commodities
K^w	Set of commodities with wrapped path
A_D^*	Set of dispatch arcs with at least one vehicle dispatch
Parameters	
σ_a	Current departure time of dispatch $a \in A_D^*$
l_k	Due time at destination terminal for commodity $k \in K$
δ	Cross-docking time at break-bulk terminals
t_a	Travel time on dispatch arc $a \in A_D^*$
$a_1^k, a_2^k, ..., a_{n_k}^k$	Sequence of dispatches used to move commodity $k \in K$
$e_{a_i^k}$	Earliest departure time of commodity k on dispatch $a_i^k \in \pi_k, i = 1, ..., n_k$
τ_W	Maximum waiting time at a terminal
Decision Variables	
α_a	Earliest departure time for dispatch $a \in A_D^*$
β_a	Latest departure time for dispatch $a \in A_D^*$

First, we introduce a constraint that ensures if all dispatches occur during their dispatch windows, then no commodity waits more than τ_W at any terminal except for the destination terminals. Once a commodity arrives to its destination terminal, we consider it to be passed to inter-city operations for further processing. Figure 3.5 visualizes the waiting time limit at terminal $U2$ for a commodity that arrives $U2$ on dispatch i and leaves $U2$ on dispatch $i + 1$.

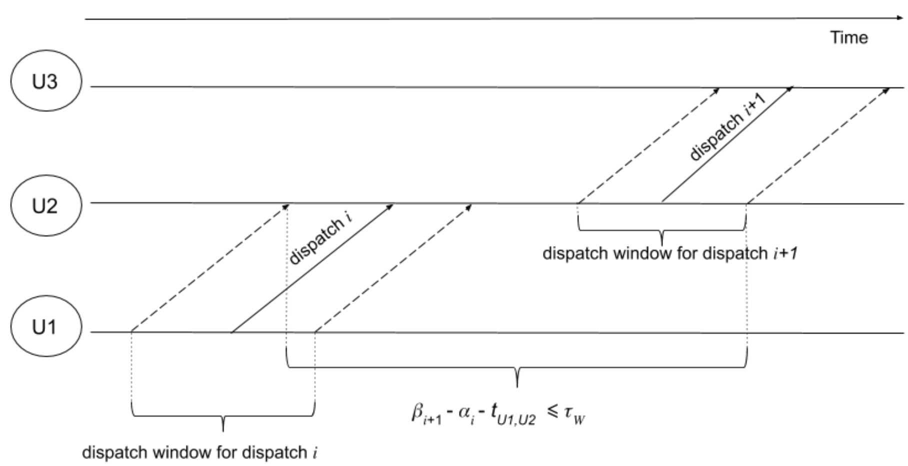


Figure 3.5: Waiting time limit at a terminal with dispatch windows

Each commodity k that visits a break-bulk terminal goes through the cross-docking process and the time it takes to cross-dock each commodity is δ. The terminals are functional during a set of operating periods within the planning period. Within each operating period, cross-docking can only start before a cut-off time that is specific to the operating period. This implies a lower bound on the earliest possible departure time for a commodity at each break-bulk terminal. Figure 3.6 illustrates the earliest departure time at break-bulk terminal, $U2$, for a commodity that arrives $U2$ on dispatch i and leaves $U2$ on dispatch $i + 1$.

We are further constrained with the dispatch window selection on a wrapped planning period. Let σ_a be departure time of a dispatch $a \in A_D$. Let K^w be set of commodities whose paths are wrapped. For each commodity $k \in K^w$, let ι_k be the index representing the dispatch that is followed by a dispatch whose arrival time is wrapped on path $\pi_k =$

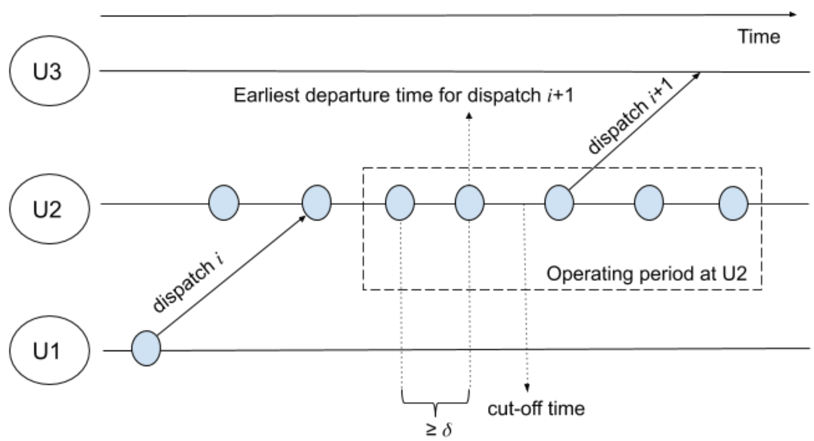


Figure 3.6: Earliest departure time for a commodity at a break-bulk terminal

$(a^k_1, ..., a^k_{n_k})$, i.e. $\iota_k < n_k$. Then, for each commodity $k \in K^w$, we take the latest possible dispatch time for dispatch $a^k_{\iota_k}$ as $\sigma_{a^k_{\iota_k}}$ and the earliest possible dispatch time for dispatch $a^k_{\iota_k+1}$ as $\sigma_{a^k_{\iota_k+1}}$. For each commodity $k \in K$, we define the earliest possible dispatch time at the origin as $e_{a^k_1} = 0$ if $\sigma_{a^k_1} < e_k$ and as $e_{a^k_1} = e_k$ if $\sigma_{a^k_1} \geq e^k$ and we assume the due time $l_k = \sigma_{a^k_{n_k}}$ if $l_k < \sigma_{a^k_{n_k}}$.

Figure 3.7 shows a wrapped commodity path. The potential dispatch windows are represented with the dashed lines.

Rather than considering individual vehicle dispatches on a dispatch arc separately, we work with dispatch arcs, A^*_D. Once we align dispatch arcs, we aggregate commodity volumes from aligned vehicle dispatches and recalculate the optimal combination of dispatch vehicles for the aggregate commodity volume.

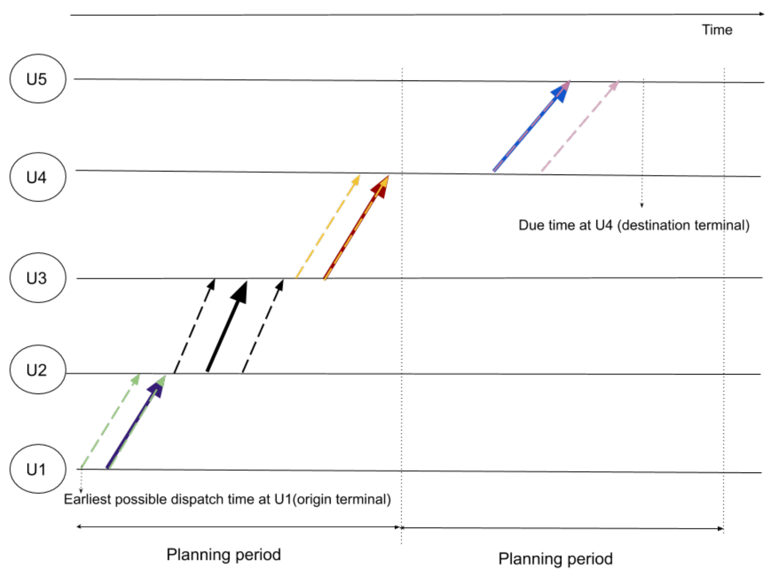


Figure 3.7: Dispatch windows with wrapped planning period

$$z^* = \max \quad \sum_{a \in A_D^*} (\beta_a - \alpha_a) \tag{3.7}$$

subject to

$$\beta_{a^k_{n_k}} + t_{a^k_{n_k}} \leq l_k, \quad \forall k \in K, a^k_{n_k} \in A_D^* \tag{3.8}$$

$$\alpha_{a^k_i} \geq e_{a^k_i}, \quad \forall k \in K, i = 1, ..., n_k, a^k_i \in A_D^* \tag{3.9}$$

$$\beta_{a^k_i} + t_{a^k_i} + \delta \leq e_{a^k_{i+1}}, \quad \forall k \in K \setminus K^w, i = 1, ..., n_k - 1, a^k_i, a^k_{i+1} \in A_D^* \tag{3.10}$$

$$\beta_{a^k_i} + t_{a^k_i} + \delta \leq e_{a^k_{i+1}}, \quad \forall k \in K^w, i = 1, ..., n_k - 1, i \neq \iota_k, a^k_i, a^k_{i+1} \in A_D^* \tag{3.11}$$

$$\beta_{a^k_{\iota_k}} = \sigma_{a^k_{\iota_k}}, \quad \forall k \in K^w, i = \iota_k, a^k_i \in A_D^* \tag{3.12}$$

$$\beta^k_{a_1} = \sigma_{a^k_1}, \quad \forall k \in K, a^k_1 \in A_D^* \tag{3.13}$$

$$\beta_{a^k_{i+1}} - \alpha_{a^k_i} - t_{a^k_i} \leq \tau_W, \quad \forall k \in K \setminus K^w, i = 1, ..., n_k - 1, a^k_i \in A_D^* \tag{3.14}$$

$$\beta_{a^k_{i+1}} - \alpha_{a^k_i} - t_{a^k_i} \leq \tau_W, \quad \forall k \in K^w, i = 1, ..., n_k - 1, i \neq \iota_k, a^k_i \in A_D^* \tag{3.15}$$

$$\alpha_{a^k_i} \leq \sigma_{a^k_i} \leq \beta_{a^k_i}, \quad \forall k \in K, i = 1, ..., n_k, a^k_i \in A_D^* \tag{3.16}$$

Constraints (3.8) ensure that each commodity arrives its destination terminal before its due time. Constraints (3.9) ensure that the earliest departure time on dispatch arc $a \in A_D^*$ is not earlier than the earliest departure of commodities carried on the dispatch arc a. Constraints (3.10)-(3.11) ensure feasible connections at break-bulk terminals. Constraints (3.12) ensure that for each commodity $k \in K^w$, the latest departure time for the dispatch with index ι_k is equal to the current departure time of dispatch, $\sigma_{a^k_{\iota_k}}$. Constraints (3.13)-(3.15) ensure that no commodity waits more than τ_W at the origin and break-bulk terminals. Constraints (3.16) ensure that the current departure time of each dispatch remains feasible. The objective is to maximize the sum of the widths of the individual windows for each dispatch arc.

There can be alternative solutions that maximize the sum of the widths of the individual windows. We select the one that maximizes the minimum width of individual windows with the following linear program.

$$\max \quad \eta \tag{3.17}$$

$$\text{subject to} \quad (3.8 - 3.16)$$

$$\sum_{a \in A_D^*} (\beta_a - \alpha_a) = z^* \tag{3.18}$$

$$\eta \leq \beta_a - \alpha_a, \quad \forall a \in A_D^* \tag{3.19}$$

Once we identify dispatch windows for each dispatch arc $a \in A_D^*$, we solve a dispatch alignment model to minimize the total number of dispatch departure times over the planning horizon. Table 3.3 summarizes the notation used in the dispatch alignment model.

Table 3.3: Notation for dispatch alignment model

Sets	
L	Set of direct lanes
A_D^{l*}	Set of dispatch arcs on lane $l \in L$
D_{a_l}	Set of time points that fall into dispatch window of $a_l \in A_D^{l*}$ on lane $l = (u_1^l, u_2^l) \in L$, i.e. $D_{a_l} \subseteq T_{u_1^l}$
Decision variables	
y_{lt}	1 if there is dispatch departure on lane $l = (u_1^l, u_2^l) \in L$ at time $t \in T_{u_1^l}$, 0 otherwise
$x_{a_l t}$	1 if dispatch vehicles on dispatch arc $a_l \in A_D^{l*}$ on lane $l = (u_1^l, u_2^l) \in L$ depart from u_1 at time $t \in D_{a_l}$, 0 otherwise

$$\min \quad \sum_{l \in L} \sum_{t \in T_{u_1^l}} y_{lt} \tag{3.20}$$

$$\text{subject to} \quad \sum_{t \in D_{a_l}} x_{a_l t} = 1, \quad \forall l \in L, \forall a \in A_D^{l*} \tag{3.21}$$

$$y_{lt} \geq x_{a_l t}, \quad \forall l \in L, \forall a_l \in A_D^{l*}, \forall t \in T_{u_1^l} \tag{3.22}$$

$$y_{lt} \in \{0, 1\}, \quad \forall l \in L, \forall t \in T_{u_1^l} \tag{3.23}$$

$$x_{a_l t} \in \{0, 1\}, \quad \forall l \in L, \forall a_l \in A_D^{l*}, \forall t \in D_{a_l} \tag{3.24}$$

Constraints (3.21) ensure that each vehicle dispatch departs at a single time point within its dispatch window. Constraints (3.22) keep track of the unique dispatch departure times. Constraints (3.23) and (3.24) define the domain of the decision variables. The objective is to minimize the sum of the number of unique dispatch departure times on the direct lanes.

We then aggregate commodities with the common departure times and identify the best combination of vehicles to carry the aggregated commodity volume with minimum transportation cost.

3.5.4 Time-expanded line-haul network refinement

The load plans generated on $TE\text{-}LN$ using optimistic and nearest mappings do not necessarily produce commodity paths that are feasible for the continuous time version of

TE-LN, i.e. having timed nodes at every minute at every terminal in the line-haul network. Both mappings underestimate the dispatch arrival times which can lead to infeasible dispatch connections and missing commodity due times at destination terminals. Pessimistic approach, on the other hand, generates feasible continuous-time solution, often with high cost.

To eliminate infeasibilities due to the discretization of time, we iteratively refine the time-expanded line-haul network by including carefully chosen time-points to TE-LN such that commodity paths that are infeasible in the fully time-expanded TE-LN, also become infeasible in the partially time-expanded TE-LN. To maintain and improve the solution quality, we also make improvement passes within the procedure. Algorithm 5 describes the steps for the time-expanded line-haul refinement, where ς controls the degree of the refinement in each iteration.

Algorithm 5: Time-expanded line-haul network refinement

while *a continuous-time feasible solution is not found* **do**

Sort commodities in descending order of the number of infeasible dispatch connections on their paths. In case of ties, sort the commodities in descending order of commodity volume.

Choose the top $\varsigma\%$ of infeasible commodity paths. Remove the chosen infeasible commodities from their paths.

Add timed nodes representing the actual dispatch arrival times on the chosen infeasible commodity paths to TE-LN.

Add start time of commodities whose dispatch times at the origin terminal is earlier than their actual earliest possible dispatch time the origin terminal.

Remove commodities whose paths were affected with the introduction of new timed nodes.

Sort all the removed commodities (K_R) in the increasing order of *routing flexibility*. In case of ties, sort the commodities in descending order of volume.

for $k \in K_R$ **do**

Select the least-cost-time-feasible path for commodity k.

end

end

3.6 Computational Study

We tested our solution approach for one of the largest Chinese small parcel carriers, SF Express. We used their data from South China from March 27, 2019 to April 1, 2019 on 59 terminal network for 60,073 commodities. Table 3.4 gives the summary statistics on the input data set and Table 3.5 lists the nine vehicle types operating in the network. SF provided travel information for 3,306 lanes which means the travel information (distance and travel time) was not available for 5% of the lanes. Among these 3,306 lanes, only

1,778 lanes have cost (per km) information for some subset of the nine vehicles and usable in our study. For each of the 60,073 commodities, the travel and vehicle information was available for the direct lane between the origin and destination terminals.

Table 3.4: Description of input data

Statistics	Value
Number of commodities	60,073
Number of terminals	59
Number of cities	28
Number of terminals (commodity origin)	58
Number of terminals (commodity destination)	55
Minimum/maximum/average time between commodity due time and release time (h)	1.0 / 71.72 / 9.05
Minimum/maximum/average commodity routing flexibility	0.09 /71.18/ 5.91

Table 3.5: Vehicle types and capacities

Type	1	1.5	2	3	5	7	14	20	30
Capacity (kg)	1,000	1,500	1,800	2,000	3,500	6,400	7,100	8,500	13,300

Figure 3.8 shows the histogram of routing flexibility for 60,073 commodities. The majority of commodities have less than 10 hours of slack time on the terminal network.

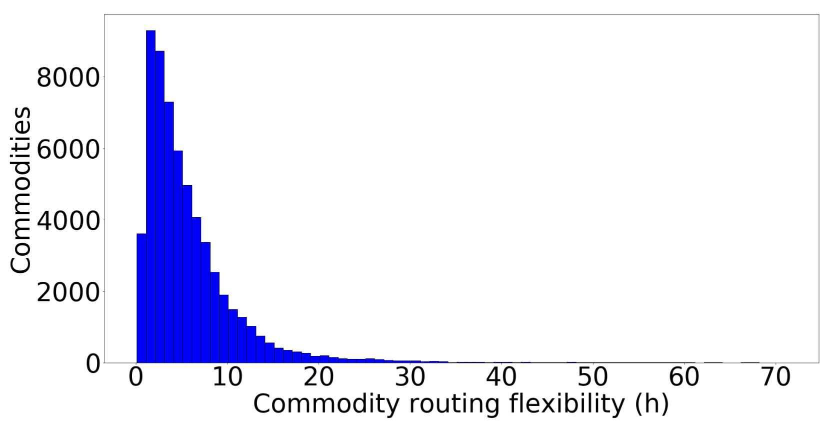


Figure 3.8: Routing flexibility

We chose a base model setting to experiment with different modeling complexities and to test different algorithmic ideas. All else being equal, we changed one or a group of parameters from the base model in our experiments. Table 3.6 shows the modeling complexities and algorithmic choices chosen in the base model setting and Table 3.7 reports the total cost after each step of the algorithm. The total cost after the iterative time refinement (ITR) denotes the total cost for the solution that is feasible for the continuous time version of the problem.

Table 3.6: Base model setting - flow planning

Modeling choices	
Operating hours at a terminal	Terminal-specific periods
Intermediate terminals	One terminal per city
Holding limit at a terminal	60 min
Cross-docking time	20 min
Algorithmic choices	
Initial discretization	20 min
Time mapping	Optimistic
% commodities routed randomly at the beginning	0%
Number of improvement passes	5 passes
Consolidation	None
% of infeasible commodities re-routed at each refinement step	50%

Table 3.7: Base model setting - total cost

Total cost	
Construction (C)	7,127,425
Improvement Passes (IP)	6,796,950
Iterative Time Refinement (ITR)	8,435,694

For each intermediate terminal in the base model setting, we calculated the commodity traffic on them and identified the following three subsets of intermediate terminals to experiment with.

- Subset 1: One terminal in the cities that have multiple terminals and a set of central terminals that were visited by more than 1,000 commodities.
- Subset 2: Terminals that were visited by more than 1,000 commodities.
- Subset 3: Terminals that were visited by more than 500 commodities.

Table 3.8 shows the selected intermediate terminals in each subset.

We used the computational cluster maintained by the School of Industrial and Systems Engineering at Georgia Institute of Technology to conduct our computational study. The

Table 3.8: Intermediate terminal selection

City	# terminals	Intermediate terminals (One per city)	# commodities via intermediate terminal (base experiment setting)	Subset 1	Subset 2	Subset 3
020	9	020X	4,341	020X	020X	020X
755	8	755R	7,327	755R	755R	755R
769	4	769WB	14,380	769WB	769WB	769WB
791	4	791W	678	791W	-	791W
757	3	757WH	2,804	757WH	757WH	757WH
595	3	595X	1,652	595X	595X	595X
591	2	591R	100	591R	-	-
592	2	592W	3,739	592W	592W	592W
752	2	752WH	482	752WH	-	-
756	2	756R	32	756R	-	-
663	2	663WH	0	663WH	-	-
797	2	797W	880	797W	-	797W
594	1	594W	778	-	-	594W
596	1	596VA	230	-	-	-
597	1	597VA	82	-	-	-
599	1	599VA	36	-	-	-
660	1	660VW	179	-	-	-
668	1	668VA	865	-	-	668VA
701	1	701VA	64	-	-	-
750	1	750W	3,253	750W	750W	750W
751	1	751VA	219	-	-	-
759	1	759VA	123	-	-	-
760	1	760W	6,991	760W	760W	760W
762	1	762VA	514	-	-	762VA
793	1	793VA	22	-	-	-
795	1	795VA	389	-	-	-
8981	1	8981V	9	-	-	-
898	1	898W	210	-	-	-

heuristic algorithm and optimization models were implemented in Python 2.7. The optimization models were solved by Gurobi 9.0. The computers used in the study had Xeon E5645 processors and 96 GB of RAM.

Table 3.9 shows the output of the experiments on modeling choices. Tables 3.10 - 3.11 give the output of the experiments on algorithmic choices. In both tables, **I** denotes the number of refinement iterations. Unless otherwise stated, the total cost refers the total cost obtained at the end of the iterative refinement step in the discussions given in Section 3.6.1 - 3.6.2.

3.6.1 Modeling choices

In this section, we discuss the experiment results, summarized in Table 3.9, for the real-life complexities we have included in our model. To begin with, the experiments with intermediate terminals show that in the absence of intermediate terminals, the total cost is almost four times as much compared to the base model setting, where one intermediate terminal is chosen for each city. Among the experiments with three subsets of intermediate terminals, the experiment with Subset 3 results in the lowest cost, which is higher than the cost of base model setting. The results stress the importance of choosing the right number and location of intermediate terminals to take advantage of consolidation opportunities. Second, we observe that the total cost is lower when terminals do not have terminal specific operating periods. However, the experiment takes almost 6 times longer to correct the infeasibilities in the solution due to time discretization. Third, the experiments with cross-docking show that shorter cross-docking time at terminals yields lower total cost. The results suggest that with longer the cross-docking times in the presence of terminal-specific operating periods, it becomes hard to find consolidation opportunities. Finally, Table 3.9 shows that the total cost ranges from 5.7 million to 50 million as the holding limit at a terminal decreases from infinity to zero. When the commodities cannot wait at terminals (Experiment: Holding limit - 0 min), the only option is to route all commodities on their direct paths, leaving the origin terminal at the earliest possible dispatch time. While this leaves no room for consolidation, having commodities wait up to an hour at their origin terminals without the option of visiting intermediate terminals (Experiment: Intermediate terminals - None) allows consolidation of commodities (with the common destination terminal) at origin terminals and decreases the total cost by almost 20 million and increases the total utilization by 2% compared to the case of no holding at a terminal (Experiment: Holding limit - 0 min). In addition to that, in the experiment with cross-docking time of 60 minutes (Experiment: Cross-docking time - 60 min), commodities can visit intermediate terminals only if they can be cross-docked upon arriving and be dispatched immediately after their cross-docking is complete (as cross-docking time is included within the holding time of 60 minutes) and this results in a total cost (23 million) lower than the total cost for the experiment with no intermediate terminal visit (Experiment: Intermediate terminals - None).

Table 3.9: Experiments on modeling complexity

	Total cost			Utilization			# Nodes		# Arcs		I	Run time(h)
	C	IP	ITR	C	IP	ITR	before ITR	after ITR	before ITR	after ITR	ITR	Total (C/IP/ITR)
Intermediate terminals												
None	27,189,369	26,198,305	30,810,429	5.99%	6.22%	5.29%	12,744	14,161	69,421	87,157	8	0.04 (0.01/0.02/0.01)
One per city	7,127,425	6,796,950	8,435,694	31.61%	32.81%	25.25%	12,744	23,984	117,950	340,057	24	2.74 (0.10/0.33/2.31)
Subset 1	7,745,302	7,343,586	9,069,066	27.96%	29.14%	22.83%	12,744	23,005	106,961	307,418	26	1.49 (0.05/0.17/1.27)
Subset 2	8,272,036	7,837,239	9,732,446	25.52%	26.74%	20.79%	12,744	22,247	94,550	270,511	27	1.26 (0.04/0.12/1.10)
Subset 3	7,559,050	7,173,985	8,871,132	29.05%	30.23%	23.69%	12,744	23,237	102,598	299,217	23	2.74 (0.10/0.32/2.31)
Operating hours at a terminal												
7/24	5,693,434	5,462,163	6,615,099	42.67%	43.86%	34.55%	12,744	30,567	152,647	596,003	30	12.98 (0.26/0.73/11.99)
Terminal-specific	7,127,425	6,796,950	8,435,694	31.61%	32.81%	25.25%	12,744	23,984	117,950	340,057	24	2.74 (0.10/0.33/2.31)
Cross-docking at a terminal												
20 min	7,127,425	6,796,950	8,435,694	31.61%	32.81%	25.25%	12,744	23,984	117,950	340,057	24	2.74 (0.10/0.33/2.31)
40 min	8,186,365	7,853,747	10,666,052	26.34%	27.23%	18.98%	12,744	23,521	113,315	308,403	28	1.67 (0.08/0.28/1.30)
60 min	10,399,532	9,904,679	23,082,097	19.33%	20.01%	7.31%	12,744	25,251	109,788	274,209	25	0.80 (0.07/0.25/0.48)
Holding limit at a terminal												
0 min	50,038,038	50,038,038	50,038,038	3.29%	3.29%	3.29%	12,744	14,064	48,843	68,808	1	0.05 (0.01/0.04/0.01)
60 min	7,127,425	6,796,950	8,435,694	31.61%	32.81%	25.25%	12,744	23,984	117,950	340,057	24	2.74 (0.10/0.33/2.31)
120 min	5,783,522	5,593,788	6,602,451	37.70%	38.60%	31.33%	12,744	22,104	131,072	340,776	21	3.68 (0.14/0.45/3.09)
180 min	5,308,635	5,151,258	5,980,601	40.33%	41.24%	34.48%	12,744	21,396	139,067	349,080	23	4.73 (0.22/0.60/3.91)
No limit	4,935,334	4,823,543	5,672,352	41.97%	42.77%	35.19%	12,744	20,536	159,737	372,421	23	8.35 (0.24/1.27/6.85)

3.6.2 Algorithmic choices

Next, we discuss the experiment results on algorithmic choices and their impact on the solution quality and computational speed. The summary results for the experiments on algorithmic choices are given in Tables 3.10 and 3.11.

The experiments on time discretization in Table 3.10 show that more granular the time-extended network, lower the total cost. The decrease in the total cost between consecutive time discretizations suggests that adding time points only yields marginal improvement in total cost while increasing run time as the time-extended network becomes more granular. For instance, while the improvement in the total cost from experiment with 40 minutes time discretization (Experiment: Discretization - 40 min) to the experiment with 20 minutes time discretization (Experiment: Discretization - 20 min) was 6.21%, it was only 0.16% from the experiment with 10 minutes time discretization (Experiment: Discretization - 10 min) to 5 minutes time discretization (Experiment: Discretization - 5 min).

In Table 3.10, the experiment with optimistic time mapping (Experiment: Time Mapping - Optimistic) results in the lowest cost compared to the nearest (Experiment: Time Mapping - Nearest) and the pessimistic (Experiment: Time Mapping - Pessimistic) time mappings and the longest run time among all mappings. Most of this run time was spent in iterative refinement step to correct infeasibilities caused by optimistic mappings of event times in the partially time-extended network.

According to our experiments, the initialization of the time-extended network by randomly routing commodities at the beginning did not have a clear impact on the total cost as show in Table 3.10.

The experiments on the number of improvement passes in Table 3.10 show that without consolidation ideas, the total cost after IP does not improve after 20 passes. With 100 improvement passes, the experiment takes 13.85 hours whereas the experiment with 5 improvement passes takes only 2.74 hours with slightly higher total cost after improvement passes. Figure 3.10 plots the total cost after each improvement pass using the the base model setting.

In Tables 3.10 - 3.11, we present the experiments on the model parameters used in Section 3.5.2 for vehicle elimination by vehicle utilization. We explore several thresholds for utilization and distance to choose dispatch arcs for vehicle removal trials. Among the experiments we performed using the base model setting, the total cost after IP is the smallest for 1000 trials with the distance threshold of 50 km and the utilization threshold of 20%. Figure 3.9 shows the change in total cost with vehicle removal for the chosen parameters at each improvement pass.

Using 1000 trials / 50km / 20% setting, we conducted experiments to understand the effect of consolidation ideas on the total cost. Compared to no consolidation, both consolidation ideas (vehicle removal and dispatch alignment) lowers total cost after improvement passes (IP). The lowest total cost after IP is obtained by combining the both consolidation ideas, i.e. performing vehicle removal trials followed by dispatch alignment at each improvement pass. However, this reduction in the total cost disappears when we correct for infeasibilities in iterative time refinement step.

One of the reason for this limited benefit derived from consolidation ideas could be the commodities with low *routing flexibility*. For instance, the attempts to consolidate commodities by vehicle removal will likely fail as the commodities with low routing flexibility are less likely to change their current paths than commodities with high routing flexibility. To investigate this further, we ran experiments (using the base model setting) only with the commodities whose maximum routing flexibility is less than 1 hour, 2 hours and 3 hours. In Table 3.12, the experiment results suggest that planning commodities with maximum routing flexibility of 1 hour, 2 hours and 3 hours (5.13%, 20.31%, 35.03% of the total commodities) account for 18.61%, 40.53%, 58.66% of the total cost of planning all commodities respectively.

The iterative time refinement strategy of correcting the half of the infeasible commodity paths at each iteration resulted in the the lowest total cost compared to correcting less than half or all of the infeasible commodity paths. Figure 3.11 shows the changes in the total cost, the number of infeasible commodities, the number of nodes and the number of arcs at each refinement iteration for the base model setting.

We also tested several scenarios to include improvement passes after each refinement iteration. The results in Table 3.11 suggest that including improvement passes in the iterative refinement step improves the total cost. We observe that with and without the consolidation ideas, the improvement passes after each iterative time refinement decreases the total cost. However, the iterative refinement step takes longer as we make improvement passes on the growing $TE\text{-}LN$ after each refinement iteration.

Table 3.10: Experiments on algorithmic choices

	Total cost			Utilization			# Nodes		# Arcs		I	Run time(h)
	C	IP	ITR	C	IP	ITR	before ITR	after ITR	before ITR	after ITR	ITR	Total (C/IP/ITR)
Discretization												
5 min	7,832,026	7,544,317	8,143,646	28.47%	29.60%	26.45%	50,976	57,531	437,267	585,911	21	4.66 (0.39/1.22/3.05)
10 min	7,534,814	7,248,570	8,156,566	29.69%	30.77%	26.14%	25,488	33,597	225,725	390,622	20	4.07 (0.19/0.71/3.17)
20 min	7,127,425	6,796,950	8,435,694	31.61%	32.81%	25.25%	12,744	23,984	117,950	340,057	24	2.74 (0.10/0.33/2.31)
40 min	8,706,621	8,140,942	8,994,137	25.29%	26.13%	23.52%	6,372	18,880	57,034	299,771	38	2.28 (0.05/0.15/2.08)
Time Mapping												
Optimistic	7,127,425	6,796,950	8,435,694	31.61%	32.81%	25.25%	12,744	23,984	117,950	340,057	24	2.74 (0.10/0.33/2.31)
Nearest	7,591,195	7,305,454	8,536,141	29.01%	29.96%	24.97%	12,744	20,226	116,271	268,342	16	1.62 (0.09/0.30/1.23)
Pessimistic	8,074,945	7,827,145	9,038,821	26.50%	27.32%	23.92%	12,744	13,983	115,754	146,672	1	0.45 (0.09/0.28/0.08)
% commodities routed randomly at the beginning - Holding limit at a terminal for path generation												
None	7,127,425	6,796,950	8,435,694	31.61%	32.81%	25.25%	12,744	23,984	117,950	340,057	24	2.74 (0.10/0.33/2.31)
5% - 0 min	7,567,457	7,154,694	8,503,264	30.05%	31.40%	25.05%	12,744	23,791	116,613	334,736	30	2.44 (0.09/0.31/2.03)
5% - 20 min	7,275,704	6,804,120	8,440,844	32.14%	33.74%	25.22%	12,744	24,184	116,494	344,550	28	3.32 (0.19/0.62/2.51)
5% - 40 min	7,371,646	6,837,822	8,330,076	31.35%	33.18%	25.60%	12,744	24,167	116,538	342,921	30	4.49 (0.24/0.57/3.67)
5% - 60 min	7,416,506	6,859,297	8,397,657	31.45%	33.28%	25.43%	12,744	24,326	116,851	346,273	29	3.26 (0.62/0.57/2.07)
50% - 20 min	19,286,899	13,696,922	8,632,831	15.52%	18.88%	24.90%	12,744	26,312	107,953	390,782	23	3.32 (0.17/0.45/2.70)
50% - 0 min	17,677,695	14,969,372	9,501,929	12.05%	14.44%	22.19%	12,744	23,927	112,266	335,808	23	2.43 (0.06/0.24/2.13)
50% - 20 min	19,286,899	13,696,922	8,632,831	15.52%	18.88%	24.90%	12,744	26,312	107,953	390,782	23	3.32 (0.17/0.45/2.70)
50% - 40 min	18,546,142	12,753,833	8,414,984	16.50%	21.05%	25.61%	12,744	26,650	109,608	397,733	21	3.79 (1.24/0.24/2.31)
50% - 60 min	18,150,819	12,252,323	8,392,383	16.63%	22.14%	25.72%	12,744	26,377	110,380	391,746	24	9.49 (6.97/0.26/2.26)
Number of improvement passes												
5	7,127,425	6,796,950	8,435,694	31.61%	32.81%	25.25%	12,744	23,984	117,950	340,057	24	2.74 (0.10/0.33/2.31)
10	7,127,425	6,792,042	8,490,094	31.61%	32.82%	25.18%	12,744	23,941	117,979	340,447	25	5.14 (0.18/1.13/3.83)
20	7,127,425	6,791,824	8,509,752	31.61%	32.82%	25.16%	12,744	23,889	117,980	338,574	25	5.84 (0.18/2.10/3.57)
40	7,127,425	6,791,824	8,509,752	31.61%	32.82%	25.16%	12,744	23,889	117,980	338,574	25	7.94 (0.18/4.17/3.60)
60	7,127,425	6,791,824	8,509,752	31.61%	32.82%	25.16%	12,744	23,889	117,980	338,574	25	10.25 (0.18/6.23/3.84)
80	7,127,425	6,791,824	8,509,752	31.61%	32.82%	25.16%	12,744	23,889	117,980	338,574	25	11.32 (0.19/7.23/3.90)
100	7,127,425	6,791,824	8,509,752	31.61%	32.82%	25.16%	12,744	23,889	117,980	338,574	25	13.85 (0.17/9.88/3.80)
Vehicle removal (Number of trials/distance threshold/utilization threshold)												
250 / 50 / 5%	7,127,425	6,514,310	8,569,390	31.61%	34.77%	24.77%	12,744	24,009	118,150	340,924	24	4.93 (0.18/0.72/4.03)
250 / 50 / 10%	7,127,425	6,448,986	8,459,400	31.61%	35.16%	25.13%	12,744	23,995	118,250	340,008	30	4.75 (0.18/0.72/3.85)
250 / 50 / 20%	7,127,425	6,426,903	8,451,404	31.61%	35.41%	25.26%	12,744	23,876	118,313	337,988	26	4.76 (0.17/0.76/3.83)
250 / 100 / 5%	7,127,425	6,468,173	8,496,096	31.61%	35.19%	25.02%	12,744	23,984	118,094	339,395	24	4.70 (0.18/0.69/3.83)
250 / 100 / 10%	7,127,425	6,434,555	8,520,622	31.61%	35.46%	24.99%	12,744	23,784	118,216	335,678	24	4.79 (0.18/0.71/3.90)
250 / 100 / 20%	7,127,425	6,384,943	8,489,035	31.61%	35.84%	25.26%	12,744	23,967	118,231	339,425	26	5.03 (0.18/0.74/4.10)
250 / 200 / 5%	7,127,425	6,482,101	8,446,882	31.61%	35.24%	25.12%	12,744	23,927	118,062	339,547	28	4.74 (0.17/0.72/3.84)
250 / 200 / 10%	7,127,425	6,396,889	8,446,245	31.61%	36.04%	25.19%	12,744	23,901	118,196	339,596	24	5.01 (0.17/0.72/4.11)
250 / 200 / 20%	7,127,425	6,342,694	8,471,956	31.61%	36.57%	24.94%	12,744	23,885	118,358	339,708	23	5.02 (0.18/0.77/4.07)

Table 3.11: Experiments on algorithmic choices (continued)

	Total cost			Utilization			# Nodes		# Arcs		I	Run time(h)
	C	IP	ITR	C	IP	ITR	before ITR	after ITR	before ITR	after ITR	ITR	Total (C/IP/ITR)
Vehicle removal (Number of trials/distance threshold/utilization threshold)												
500 / 50km / 5%	7,127,425	6,423,630	8,442,644	31.61%	35.46%	25.27%	12,744	23,843	118,282	338,168	22	5.62 (0.20/0.98/4.44)
500 / 50km / 10%	7,127,425	6,310,806	8,478,702	31.61%	36.18%	25.20%	12,744	23,892	118,346	338,899	23	2.86 (0.10/0.51/2.25)
500 / 50km / 20%	7,127,425	6,241,896	8,517,316	31.61%	36.80%	25.09%	12,744	23,838	118,504	337,777	22	4.84 (0.18/0.91/3.74)
500 / 100km / 5%	7,127,425	6,408,635	8,349,937	31.61%	35.63%	25.73%	12,744	23,935	118,147	337,463	28	4.79 (0.18/0.90/3.70)
500 / 100km / 10%	7,127,425	6,291,637	8,418,735	31.61%	36.57%	25.50%	12,744	23,975	118,407	340,388	25	4.96 (0.18/0.91/3.87)
500 / 100km / 20%	7,127,425	6,189,920	8,543,643	31.61%	37.40%	24.92%	12,744	23,861	118,399	338,388	31	4.82 (0.18/0.92/3.71)
500 / 200km / 5%	7,127,425	6,462,729	8,460,887	31.61%	35.44%	25.20%	12,744	23,933	118,109	338,574	24	4.78 (0.18/0.89/3.71)
500 / 200km / 10%	7,127,425	6,351,417	8,503,996	31.61%	36.37%	25.15%	12,744	23,889	118,272	335,844	26	4.98 (0.17/0.90/3.91)
500 / 200km / 20%	7,127,425	6,269,219	8,498,385	31.61%	37.30%	25.11%	12,744	24,026	118,337	340,760	29	5.07 (0.18/0.91/3.99)
1000 / 50km / 5%	7,127,425	6,374,155	8,492,010	31.61%	35.74%	25.05%	12,744	23,942	118,277	338,773	28	5.07 (0.16/1.10/3.81)
1000 / 50km / 10%	7,127,425	6,216,909	8,510,332	31.61%	37.03%	25.11%	12,744	23,839	118,489	337,781	22	4.44 (0.17/1.12/3.15)
1000 / 50km / 20%	7,127,425	6,051,974	8,451,834	31.61%	38.24%	25.23%	12,744	23,910	118,628	340,858	31	4.96 (0.16/1.10/3.70)
1000 / 100km / 5%	7,127,425	6,385,511	8,411,663	31.61%	35.76%	25.43%	12,744	24,000	118,183	339,233	24	5.42 (0.19/1.22/4.01)
1000 / 100km / 10%	7,127,425	6,240,619	8,488,872	31.61%	36.98%	25.17%	12,744	23,852	118,409	335,736	24	4.81 (0.17/1.15/3.49)
1000 / 100km / 20%	7,127,425	6,089,631	8,368,735	31.61%	38.18%	25.49%	12,744	23,835	118,526	338,834	24	4.29 (0.18/1.19/2.92)
1000 / 200km / 5%	7,127,425	6,467,119	8,514,460	31.61%	35.44%	25.05%	12,744	23,887	118,110	336,219	22	5.42 (0.19/1.19/4.03)
1000 / 200km / 10%	7,127,425	6,349,825	8,500,688	31.61%	36.37%	25.04%	12,744	23,924	118,232	340,906	27	3.02 (0.10/0.72/2.21)
1000 / 200km / 20%	7,127,425	6,249,077	8,481,045	31.61%	37.48%	25.14%	12,744	24,016	118,380	339,381	27	5.39 (0.20/1.24/3.95)
Consolidation												
No consolidation	7,127,425	6,796,950	8,435,694	31.61%	32.81%	25.25%	12,744	23,984	117,950	340,057	24	2.74 (0.10/0.33/2.31)
Vehicle removal (1000/50km/20%)	7,127,425	6,051,974	8,451,834	31.61%	38.24%	25.23%	12,744	23,910	118,628	340,858	31	4.96 (0.16/1.10/3.70)
Dispatch windows	7,127,425	6,739,272	8,452,326	31.61%	33.05%	25.32%	12,744	23,766	118,302	335,702	23	2.67 (0.10/0.53/2.04)
Both	7,127,425	6,031,507	8,420,769	31.61%	38.25%	25.30%	12,744	23,928	119,008	338,768	23	3.13 (0.09/0.81/2.23)
% of infeasible commodities rerouted at each refinement step												
25%	7,127,425	6,796,950	8,580,273	31.61%	32.81%	24.67%	12,744	23,214	117,950	326,250	43	4.78 (0.18/0.59/4.02)
50%	7,127,425	6,796,950	8,435,694	31.61%	32.81%	25.25%	12,744	23,984	117,950	340,057	24	2.74 (0.10/0.33/2.31)
75%	7,127,425	6,796,950	8,484,611	31.61%	32.81%	25.19%	12,744	24,308	117,950	344,517	15	2.95 (0.10/0.34/2.50)
100	7,127,425	6,796,950	8,505,362	31.61%	32.81%	24.99%	12,744	24,898	117,950	356,993	16	2.62 (0.10/0.33/2.20)
Consolidation - Number of passes in improvement step - Number of improvement passes after each refinement step												
No consolidation / 5 / 0	7,127,425	6,796,950	8,435,694	31.61%	32.81%	25.25%	12,744	23,984	117,950	340,057	24	2.74 (0.10/0.33/2.31)
No consolidation / 5 / 5	7,127,425	6,796,950	7,966,850	31.61%	32.81%	27.34%	12,744	24,634	117,950	362,759	30	53.90 (0.10/0.32/53.49)
Vehicle removal / 5 / 0 (1000 / 50km / 20%)	7,127,425	6,051,974	8,451,834	31.61%	38.24%	25.23%	12,744	23,910	118,628	340,858	31	4.96 (0.16/1.10/3.70)
Vehicle removal / 5 / 5 (1000 / 50km / 20%)	7,127,425	6,051,974	8,072,511	31.61%	38.24%	26.74%	12,744	24,751	118,628	364,690	28	42.56 (0.08/0.61/41.86)
Dispatch windows / 5 / 0	7,127,425	6,739,272	8,452,326	31.61%	33.05%	25.32%	12,744	23,766	118,302	335,702	23	2.67 (0.10/0.53/2.04)
Dispatch windows / 5 / 5	7,127,425	6,739,272	8,058,334	31.61%	33.05%	27.00%	12,744	24,509	118,302	357,862	24	39.38 (0.10/0.39/38.89)
Both / 5 / 0	7,127,425	6,031,507	8,420,769	31.61%	38.25%	25.30%	12,744	23,928	119,008	338,768	23	3.13 (0.09/0.81/2.23)
Both / 5 / 5	7,127,425	6,031,507	8,016,860	31.61%	38.25%	27.04%	12,744	24,491	119,008	357,551	30	52.54 (0.10/0.79/51.66)

Table 3.12: Commodity maximum routing flexibility(h)

	# commodities	Total cost			Utilization			# Nodes		# Arcs		I	Run time(h)
		C	IP	ITR	C	IP	ITR	before ITR	after ITR	before ITR	after ITR	ITR	Total (C/IP/ITR)
Maximum routing flexibility (h)													
1	3,081	1,374,574	1,325,198	1,569,664	3.39%	3.53%	2.92%	12,312	13,074	27,635	32,295	8	0.03 (0.01/0.01/0.01)
2	12,198	2,905,954	2,805,186	3,418,946	12.69%	13.28%	10.53%	12,312	14,771	50,771	75,447	14	0.05 (0.01/0.02/0.02)
3	21,041	4,219,137	3,992,379	4,948,685	17.40%	18.65%	14.33%	12,528	16,595	64,837	117,086	16	0.11 (0.02/0.04/0.06)
∞	60,073	7,127,425	6,796,950	8,435,694	31.61%	32.81%	25.25%	12,744	23,984	117,950	340,057	24	2.74 (0.10/0.33/2.31)

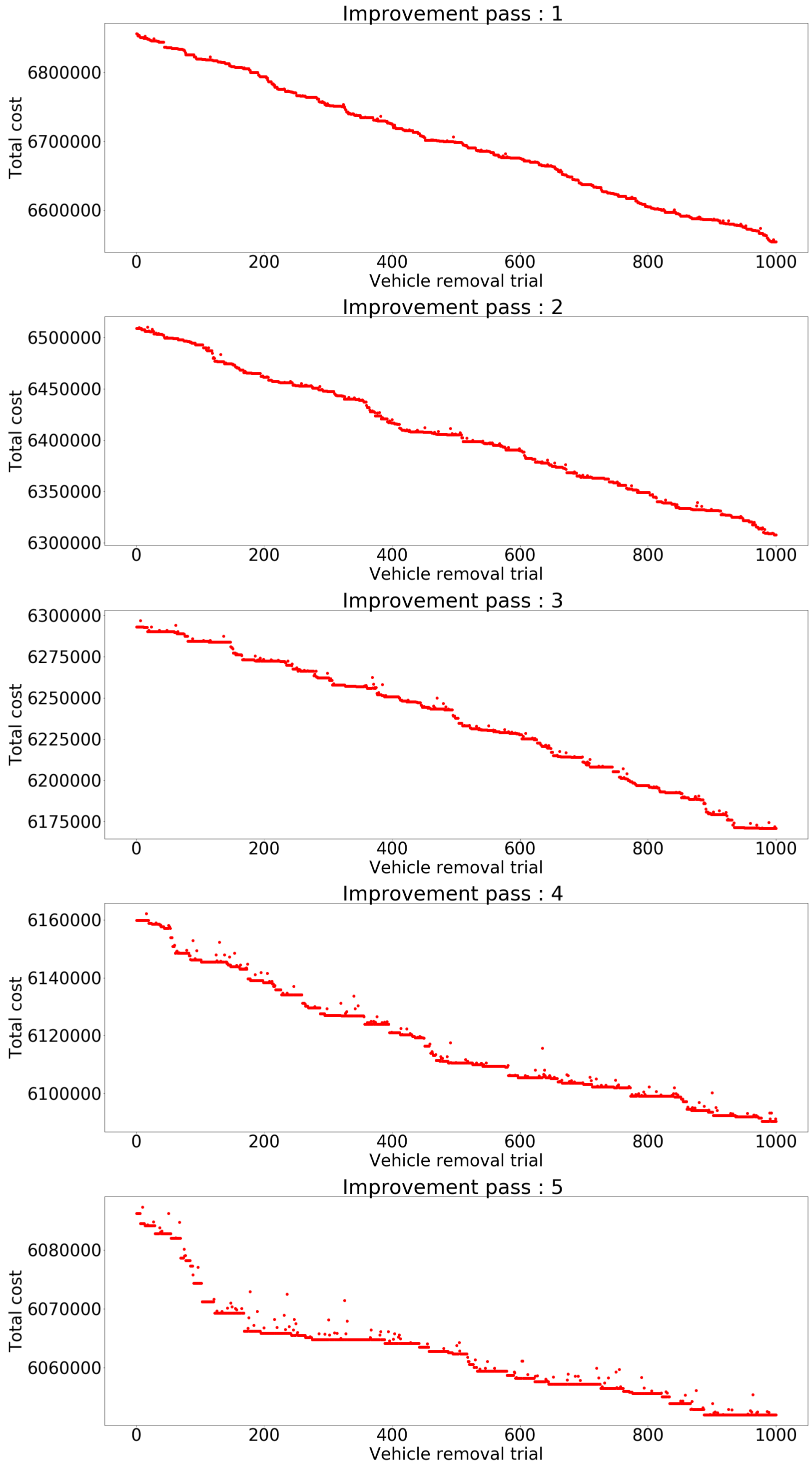


Figure 3.9: Vehicle removal trails

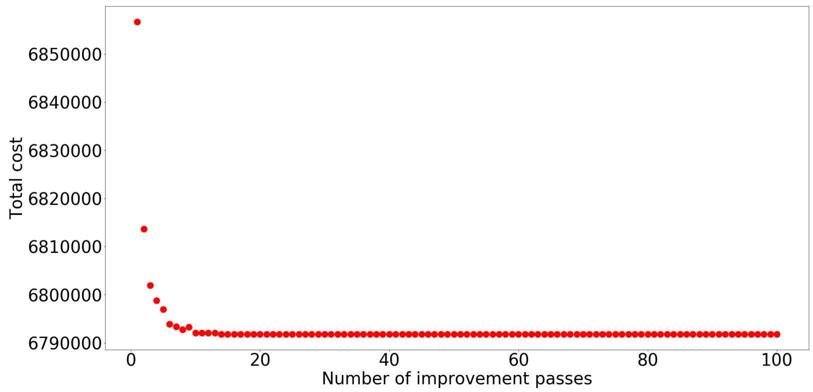


Figure 3.10: Improvement passes

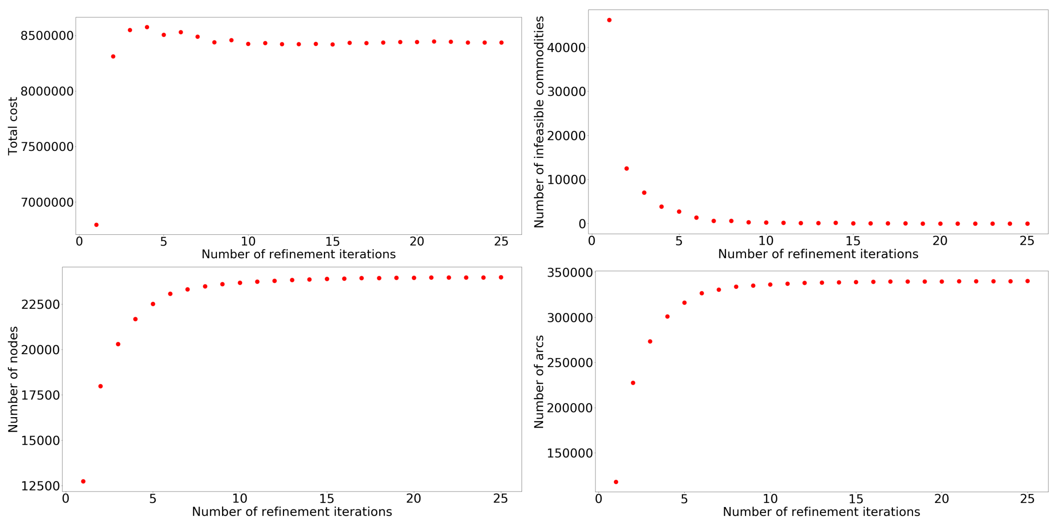


Figure 3.11: Iterative time refinement

3.6.3 Terminal utilization

We further investigate how the commodities utilize terminals and how the algorithmic and modeling choices affect the frequency of visits to terminals using the base model setting.

Intermediate terminal visits

The experiments in Table 3.9 show the benefit of having intermediate terminals to consolidate commodities at central locations and to decrease the total transportation cost. Depending on their routing flexibility, the commodities utilize intermediate terminals in different levels. In Table 3.13, we summarize the number of commodities, the average routing flex-

Table 3.13: Intermediate terminals on commodity paths

# intermediate terminals	# commodities	Average direct distance(km)	Average direct travel time(h)
0	25,047	126.21	1.84
1	18,055	224.25	3.09
2	10,153	336.93	4.52
3	4,491	430.34	5.75
4	1,637	492.84	6.62
5	514	496.23	6.74
6	131	501.08	6.75
7	36	510.74	6.75
8	7	513.02	7.12
9	2	462.75	6.04

ibility, the average distance and travel time on direct lane between origin and destination terminals broken down by the number of intermediate terminals visited on a commodity path. The results show that for a few commodities that visited 5 or more intermediate terminals, the average direct distance is more than twice the average distance for commodities that are routed on their direct lane between origin and destination terminals.

Cycling between terminals

In the load plan generated using the base model setting, we observe that 5.48% of the total commodities visit at least one terminal (except for its destination terminal) more than once as shown in Figure 3.12.

One possible explanation to these multiple visits can be that by cycling between terminals, commodities can avoid the holding time limit at terminals (60 minutes in the base model setting) and align with commodities in time for further consolidation if they can be routed for free on dispatch arcs with extra capacity. To test our hypothesis, in our marginal cost calculation for the dispatch arcs (see Section 3.5.1), we introduce a fixed penalty[1] on each dispatch arc. As holding commodities at a terminal is assumed not to incur any cost, this penalty prioritizes holding over dispatch and thus deincentivizes commodities cycling

[1]We do not include this penalty in our total cost calculation.

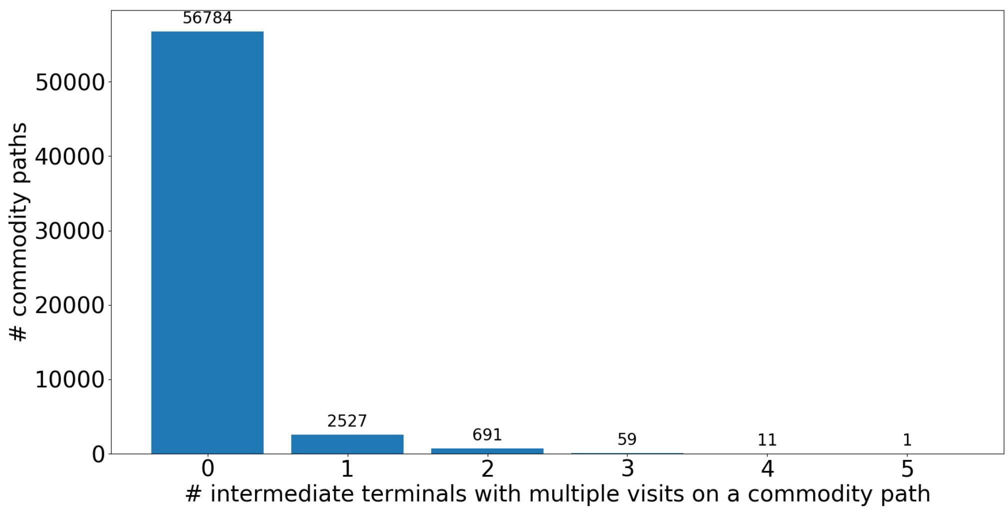


Figure 3.12: Cycling between terminals

between terminals only because it is free to do so. To understand how sensitive the cycling behaviour is to this penalty, we experimented with five penalty values ranging from 1 to 1,000. Table 3.15 summarizes the cycling analysis on the number of intermediate terminals visited more than once for the chosen penalty values and Table 3.16 summarizes the results of these experiments. Compared to no penalty case, having a positive penalty decreases the number of commodities that visit at least one terminal more than once. While the percentage of commodity paths without cycling increases from 94.52% to 99.86%, the total cost also increased from 8,435,694 to 17,311,570 as the penalty increases from 0 to 1000. As the penalty increases, we see that the number of commodities routed on their direct lanes between their origin and destination terminals also increases (Table 3.14).

Table 3.14: Commodities on direct lanes

Cycling penalty	0	1	10	100	1000
# commodities	25,047	27,091	27,527	30,523	50,254

Table 3.15: Cycling penalty analysis - intermediate terminals

# intermediate terminals visited more than once	Cycling penalty				
	0	1	10	100	1000
0	56,784	58,770	58,760	59,304	59,998
1	2,527	1,144	1,164	704	73
2	691	148	143	65	2
3	59	10	6	0	0
4	11	1	0	0	0
5	1	0	0	0	0
Total	60,073	60,073	60,073	60,073	60,073

Table 3.16: Experiments on cycling penalty

	Total cost			Utilization			# Nodes		# Arcs		I	Run time(h)
	C	IP	ITR	C	IP	ITR	before ITR	after ITR	before ITR	after ITR	ITR	Total (C/IP/ITR)
Cycling penalty												
0	7,127,425	6,796,950	8,435,694	31.61%	32.81%	25.25%	12,744	23,984	117,950	340,057	24	2.74 (0.10/0.33/2.31)
1	6,961,928	6,614,347	8,283,163	31.77%	31.94%	24.92%	12,744	22,924	120,717	330,584	20	6.81 (0.13/0.96/5.73)
10	7,015,352	6,680,597	8,502,781	31.32%	31.54%	24.36%	12,744	22,500	120,309	316,962	21	4.22 (0.25/1.11/2.86)
100	7,565,711	7,182,345	9,000,100	27.44%	28.06%	22.04%	12,744	21,764	120,096	299,715	30	5.53 (0.24/1.04/4.24)
1000	14,905,263	14,119,143	17,311,570	12.38%	13.09%	10.47%	12,744	17,500	120,599	207,631	16	1.17 (0.12/0.41/0.64)

To understand how sensitive the cycling behavior to the holding limit at terminals is, we ran experiments with different cycling penalties and holding limits. Table 3.17 shows the percentage of commodity paths without cycling between terminals for a set of cycling penalty and holding limit combinations. When holding limit is 0, commodities are routed on the direct lanes between their origin and destination terminals and there is no cycling. With and without the cycling penalty, we see that the percentage of commodity paths without cycling increases as the holding limit increases. The percentage of commodity paths without cycling increases as the cycling penalty increased expect for few cases. The results are less sensitive to the penalty increase from 1 to 10 compared to the penalty increases from 10 to 100 or from 100 to 1,000. As expected, without holding limit, as soon as we introduce a penalty cost, all commodity paths become free of cycling.

Table 3.17: Commodity paths without cycling

	Cycling penalty				
Holding limit	0	1	10	100	1000
0	100%	100%	100%	100%	100%
60 min	94.52 %	97.83%	97.81%	98.72%	99.86%
120 min	95.38%	98.91%	98.81%	99.23%	99.96%
180 min	96.46%	99.21%	99.46%	99.58%	99.99%
No limit	97.54%	100%	100%	100%	100%

Table 3.18: Experiments on cycling penalty and holding limit

	Total cost			Utilization			# Nodes		# Arcs		I	Run time(h)
	C	IP	ITR	C	IP	ITR	before ITR	after ITR	before ITR	after ITR	ITR	Total (C/IP/ITR)
Cycling penalty and holding limit												
0 min / 0	50,038,038	50,038,038	50,038,038	3.29%	3.29%	3.29%	12,744	14,064	48,843	68,808	1	0.05 (0.01/0.04/0.01)
0 min / 1	50,038,038	50,038,038	50,038,038	3.29%	3.29%	3.29%	12,744	14,064	48,843	68,808	1	0.11 (0.02/0.07/0.01)
60 min / 0	7,127,425	6,796,950	8,435,694	31.61%	32.81%	25.25%	12,744	23,984	117,950	340,057	24	2.74 (0.10/0.33/2.31)
60 min / 1	6,961,928	6,614,347	8,283,163	31.77%	31.94%	24.92%	12,744	22,924	120,717	330,584	20	6.81 (0.13/0.96/5.73)
120 min / 0	5,783,522	5,593,788	6,602,451	37.70%	38.60%	31.33%	12,744	22,104	131,072	340,776	21	3.68 (0.14/0.45/3.09)
120 min / 1	5,671,252	5,447,832	6,496,032	37.05%	37.20%	30.76%	12,744	20,935	137,360	329,173	22	10.26 (0.45/2.04/7.77)
180 min / 0	5,308,635	5,151,258	5,980,601	40.33%	41.24%	34.48%	12,744	21,396	139,067	349,080	23	4.73 (0.22/0.60/3.91)
180 min / 1	5,207,298	5,040,687	5,887,412	39.41%	39.44%	33.61%	12,744	20,508	147,532	342,704	23	12.11 (0.60/2.31/9.20)
No limit / 0	4,935,334	4,823,543	5,672,352	41.97%	42.77%	35.19%	12,744	20,536	159,737	372,421	23	8.35 (0.24/1.27/6.85)
No limit / 1	4,851,763	4,741,609	5,600,955	40.34%	40.46%	33.91%	12,744	19,307	207,075	394,967	26	24.35 (0.98/5.68/17.70)

3.7 Conclusion

In this chapter, we have introduced a marginal cost path-based greedy heuristic for the solution of large-scale instances of the service network design problem, i.e., the routing of freight from its origin to its destination while minimizing total transportation costs in the transportation (line-haul) network. We have also introduced a novel use of iterative time refinement within the greedy heuristic to obtain a continuous-time feasible solution of the service network design problem. Our algorithmic design choices have been motivated by the need to solve real-world instances for which it is impossible to rely on integer programming approaches (even with the most advances, fastest commercial solvers). In addition to its speed, the approach also allows the incorporation of many real-world complexities, which is necessary if optimization technology is to be used in practice. The output of the solution approach is a load plan which specifies the vehicle dispatches between terminals and the origin-destination route for each commodity (in terms of vehicle dispatches from each visited terminal). The load plan implicitly assumes that all planned dispatches are performed as one-way moves. However, this is not the reality because the drivers that perform the planned dispatches have to return home. Having drivers return home with an empty vehicle is too costly and unrealistic in practice. For this reason, in Chapter 4, we look into identifying feasible driver schedules to execute a load plan, so as to minimize the total transportation and operational costs while adhering to realistic constraints such as hours of service regulations.

CHAPTER 4
DRIVER CONSIDERATIONS AND OUTSOURCING IN SERVICE NETWORK DESIGN

4.1 Introduction

Companies in the express and parcel delivery business (e.g. UPS (US), DHL (Europe), Toll Express (Australia), SF Express (China)) offer their customers reliable and fast delivery of packages with varying sizes and weights between locations that are far apart from each other. With the rapid growth of e-commerce industry in the recent years, these companies have to deal with increased freight volume with tighter service level guarantees. In order to keep their costs down, they continuously revise their strategic and tactical plans for their line-haul transportation.

When planning line-haul operations, companies choose the mode(s) of transportation they will use to move freight within the line-haul network. Each mode of transportation requires a different set of resources and has different cost components. Along with the mode choice, companies make labor and equipment planning to determine their long-term and seasonal resource needs in order to provide reliable and fast service to their customers. In tactical level, they focus on flow planning and route optimization in order to utilize the available resources in the most cost-effective way possible.

Line-haul drivers and vehicles are two main resources for companies in ground freight transportation business. Based on line-haul demand estimates, companies maintain a pool of drivers and vehicles to ensure fast and reliable service to their customers. One way to build such a pool is to hire drivers who become either full-time or part-time company employees and to invest in company-owned vehicles. Companies can also outsource part of their transportation operations to the contractors who want to leverage their empty back-haul capacity and/or any available resources in their line-haul network. These contractors agree to reserve their drivers and vehicles to the exclusive use of the shippers following the terms of contract of carriage signed between two parties. The contract terms may specify the length of contract, the rates for different driver and vehicle types on a set of lanes over the contractual period of time.

By investing in their own fleet of drivers and vehicles, companies do not have to rely on other carriers to provide reliable service to their customers and they have more flexibility on how they can utilize their resources. However, this flexibility brings high capital and maintenance costs for company-owned vehicles, additional workload of recruiting and training prospective company drivers and high compensations offered to retain them. With outsourcing, companies can cut many of these expenses, but they accept the risk of resource shortage when they need them.

Either hired or outsourced, the line-haul drivers can work as local, regional or over the road (OTR) drivers. Local drivers operate usually alone on short routes between local terminals that are within a certain radius (e.g. 300 miles) of their home base. Thus, they can be home daily at the end of their shifts. Regional drivers haul freight within a specific region which is larger than a local driver's coverage. These drivers are usually on the road during the week and return home for the weekends. OTR drivers are long-haulers who are on the road for an extended period of time hauling freight long distances. They return home less frequently than regional drivers. Both regional and OTR drivers can work solo or team up with another driver. While the local drivers are often expected to load and unload the freight by themselves, regional and OTR drivers are usually responsible for only dropping one trailer and picking up another one at a terminal. The dock workers sort, handle, load and unload the incoming and outgoing freight during the terminal operating periods. The drivers are either paid salary or per unit distance traveled which may vary by the vehicle type and the lane. Driver may receive hourly or flat rate for layovers.

Depending on the scale of company operations and the anticipated demand surges, companies build their own pool of drivers and vehicles to move freight in their line-haul network. Each driver is assigned to a dispatch route which is composed of one or more planned dispatches and potentially empty legs. Either hired or outsourced, each route needs to comply with the regulations on working hours of drivers such as the rest breaks, the limit on consecutive driving hours and the daily limit on total driving hours.

Companies can only plan round trips for their company drivers who expect to return back home at the end of their shifts. Assigning single dispatch to a company driver means that the driver will drive back empty and still get paid for both the loaded and the empty legs. On the other hand, when companies outsource dispatches, they only pay for costs specified on the contract for the outsourced move. Contractors also need to build cycles for their drivers, but they do it by combining dispatches from multiple shippers. Sometimes, contractors are willing to offer discounts when shippers offer complete cycles. Although a contractor might be willing to serve shipper cycles, it is unlikely that a shipper outsources all of its transportation operations as it would create complete dependency to the contractors.

Table 4.1 shows an example of cost rates for driver and movement types where cycle outsourcing is the cheapest option. It costs \$296, \$243 and \$240 to ship a freight with total weight of 800 kg from A to B with company drivers, outsourcing a one-way move and outsourcing a cycle respectively.

Table 4.1: Example cost rates for driver and movement types

Lane (From - To)	Vehicle Capacity (kg)	Distance (km)	Company One-way (per km)	Contractor One-way (per km)	Contractor Cycle (per km)
A - B	5,000	90	1.6	2.7	1.4
B - A	5,000	95	1.6	2.7	1.2

In this chapter, we study the driver scheduling problem in service network design in which we look into identifying time-feasible driver schedules to execute a *load plan*, i.e., the set of planned vehicle dispatches, so as to maximize the total cost savings over the (unrealistic) scenario in which company drivers perform one-way moves and return empty while respecting hours-of-service regulations and company rules. In addition to company operations, we consider the option of outsourcing transportation and assess its benefits for different negotiated prices with contractors.

The remainder of this chapter is structured as follows: Section 4.2 explores the related literature review on the vehicle routing and driver scheduling problems. Section 4.3 states the problem with the modeling choices and assumptions. Section 4.4 describes the solution approach. Section 4.5 details how we generate input data for computational study. Section 4.6 discusses the experiments and their results. Finally, Section 4.7 summarize the main contributions and future research directions.

4.2 Literature Review

To best of our knowledge, there are few, if any, studies in the literature that address driver considerations and outsourcing opportunities in service network design, even though these are critically important in real-world settings. On the other hand, there are several studies in *vehicle routing problem (VRP)* literature that consider working hours and rest periods of truck drivers and outsourcing.

Similar to our work, [56] presented a scheduling approach which creates operational schedules that can be used to estimate the execution cost of a given load plan. Their approach first identifies time windows for each dispatch in a given load plan. Using these time windows, it creates loaded truck dispatches between terminals. Their set covering model selects a subset of driver tours which takes 2009 U.S. Hours of Service Regulations into account. The authors used column generation to solve the linear programming relaxation.

In one of the early works, [58] considered lunch breaks and night rests within *VRP*. They handled these rest periods in a branch-and-price algorithm for a pickup and delivery problem and heuristically solve large scale instances in a dynamic planning environment. [59] modeled rest breaks as fictitious customers. [60] showed that the maximum working time constraints can be incorporated as a resource constraint within constraint shortest path algorithms. Similarly, [61] introduced a modified insertion heuristic to deal with the maximum shift time limits.

To improve drivers' working conditions and ensure road safety, many governments worldwide impose hours of service regulations for long-haul freight transportation. These regulations limit the driving and working times by enforcing compulsory breaks and rest periods within and between driver shifts and noncompliance with the rules can result in considerable fines. For instance, in the United States (US), the hours of service regulations that put in motion in 2013 [62] states that i) a driver must not drive for more than 11 hours without taking a rest period of at least 10 consecutive hours, ii) a driver must not drive after 14 hours have elapsed since the end of last rest period and iii) a driver must not drive if 8 hours have elapsed since the end of the last rest or break period of at least 30 minutes. Previous regulations, which did not prohibit driving without taking a break of at least 30 minutes, allowed driving up to 11 hours without a break [63].

When planning for long-haul vehicle routes, ignoring local hours of service regulation will likely to result in infeasible driver schedules. The first work that explicitly considered hours of service regulations in vehicle routing literature was [64] who studied *VRP* with multiple time windows in the presence of US hours of service regulations. The authors presented a column generation algorithm that relies on a fast heuristic to solve the sub-problem to schedule on and off duty periods. Several other studies have proposed solution approaches to address the hours of service regulations in the United States ([65], [66],[67]), European Union ([68],[69],[70]), Canada ([71]) and Australia ([72]).

A number of studies focused on evaluation of feasible driver schedules in the presence of hours of service regulations. For instance, [65] presented a polynomial time approach to determine feasibility of a driver schedule to execute a sequence of full truckload requests considering US hours of service regulations of study's time period. [66] showed this problem can be solved in $\mathcal{O}(k^2)$ where k is the number of locations in driver route.

In *VRP* literature, [73] considered the option of outsourcing tours as part of operational planning for transportation providers including package delivery companies. They formulated the problem of vehicle routing and personnel planning as a nonlinear mixed-integer programming model. Due to the size of the real-life instances, they developed a solution approach based on tabu search algorithm.

4.3 Problem Definition

We consider the driver scheduling problem faced by parcel delivery companies to cover a set of planned dispatches in (a specific region of) their line-haul network using the drivers and the equipment at their disposal. Our study is motivated by the environment encountered at SF Express, one of the largest parcel carriers in China. We investigate the problem with the following model complexities:

1. *Hours of service regulations*: Governments typically impose many restrictions on the working hours of drivers to ensure the safety of these drivers and the general public. The restrictions are captured in the *hours of service* regulation. Regulations differ from country to country, but the main characteristics are the same. We take the following restrictions into account. When a driver has operated a vehicle for x hours, then the driver has to rest for at least y (consecutive) hours. Similarly, when a driver has been on-duty, i.e. operating a vehicle, waiting at a terminal while the vehicle is being loaded/unloaded, or simply waiting for z hours, then the driver has to rest again for at least y (consecutive) hours.

2. *Hours of operations*: Each terminal can, in principle, be operational 24 hours a day, 7 days a week. However, in practice there is typically a set of pre-defined operating periods during which a terminal is active in the planning horizon. When vehicles arrive at a terminal outside the operating periods then it has to wait until the next operating period begins (which counts towards hours on-duty and may also trigger company specific rules, e.g., a limit on waiting time at a terminal).

3. *Freight handling*: Upon arriving to a terminal, each vehicle is unloaded, unless it is an empty move. Before departing from a terminal, each vehicle is loaded, unless it is an empty move.

4. *Movement types*: There are two types of movement: *one-ways* and *cycles*. A one-way represents movement on a single lane, while a cycle is a round-trip with multiple lanes that starts and ends at the same terminal (we only consider simple cycles). A cycle with two lanes where one lane's destination is the other lane's origin is called *out-and-back*. All cycles can be one-day or two days long. All two-day cycles include a layover to respect the hours of service regulations.

5. *Driver types*: There are two types of drivers: company drivers and outsourced drivers (contract drivers or contractors). A company driver can only be assigned to round-trips that start and end at driver's domicile while outsourced drivers can perform both round-trips and one-way moves.

6. *Vehicle types*: Companies have their own fleet of vehicles with different capacities. Based on the terms of their contract, contractors can operate certain vehicle types on certain lanes in the line-haul network.

Given these modeling complexities, we make the following assumptions for driver scheduling problem.

- There are sufficiently many drivers and sufficiently many vehicles of each type $v \in V$.
- Each driver can operate each vehicle type.
- Outsourcing is available for all or for subset of the lanes.
- Outsourced cycles can only contain (both empty or loaded) lanes where outsourcing is available.

- Contractors expect one-way moves or cycles to be longer than an agreed minimum distance. We consider 50 km as the minimum distance.
- Both company and contract drivers are compensated per kilometer. The rate is different for each lane, driver type, vehicle type, and movement type. For instance, the cost per km can be different for lanes in opposite directions (depending on the driver's domicile). The cost of a round-trip is based on the total distance (i.e., loaded plus empty) and layover cost, if any.
- Each schedule starts with a loaded dispatch.
- The loading and unloading times are same for each vehicle type.
- It is possible to consider terminal specific operating periods. But for the sake of simplicity, we assume the terminals operate 24/7.
- There is a limit on the time a vehicle can wait at a terminal.
- For each mandatory rest period on a driver's schedule, there is a fixed layover cost.
- When a driver has operated a vehicle for 12 hours, then the driver has to rest for at least 9 (consecutive) hours.
- When a driver has been on-duty, i.e. operating a vehicle, waiting at a terminal while the vehicle is being loaded/unloaded, or simply waiting for 15 hours, then the driver has to rest for at least 9 (consecutive) hours.
- In a two-day cycle, the layover can only occur at locations(terminals) that are sufficiently far away (300 km) from the driver's domicile, i.e. cycle origin.

4.4 Solution Approach

Let $LN = (U, L)$ be the company's line-haul network with the node set U representing the set of terminals in the network and the arc set, L, is the set of lanes connecting terminals.

Let LP represent a load plan which specifies the set of vehicle dispatches, Θ, between terminals and origin-destination route for each commodity (in terms of vehicle dispatch from each visited terminal) in LN over a predetermined (wrapped) planning horizon. A (planned) vehicle dispatch $\vartheta \in \Theta$ is then characterized by $(o_\vartheta, dt_\vartheta, d_\vartheta, at_\vartheta, v_\vartheta)$ where $o_\vartheta \in U$ is the origin terminal, dt_ϑ is the departure time at origin terminal, $d_\vartheta \in U$ is destination terminal, at_ϑ is the arrival time at destination terminal, v_ϑ the vehicle type and K_ϑ is set of commodities carried on the dispatch ϑ.

We begin by constructing a set of time-feasible candidate cycles that covers the planned dispatches in Θ. As an unrealistic yet feasible base scenario, we consider company drivers perform one-way moves and return empty and thus cover all the planned dispatches.

4.4.1 Generating time-feasible n-cycle

We define a time-feasible n-cycle as a collection of n time-feasible dispatches, where the origin terminal of the earliest dispatch is the same as the destination terminal of the latest dispatch and it is possible to travel from the destination location of a dispatch to the original location of the subsequent dispatch in the time between the arrival at the destination location of the first dispatch and the departure from the original location of the second dispatch. Depending on the experiment parameters, a time-feasible n-cycle may contain empty travel from the destination location of a dispatch to the original location of the subsequent dispatch. The dispatches in a time-feasible cycle also have to comply with hours of service regulations (e.g. limits on total driving and on duty times) and company operating rules (e.g. layover restrictions).

We generate all the time-feasible n-cycles starting with a planned dispatch $\vartheta \in \theta$ using a modified depth-first search (DFS) algorithm. We run DFS on the time-extended network graph introduced in Chapter 3 with predetermined time discretization and all of the planned dispatches from a load plan generated in the same chapter. We track visited locations to prevent creation of sub-cycles. We track total daily driving and on-duty time. In a two-day cycle, the layover can only occur at locations (terminals) that are sufficiently far away from the driver's domicile. We account for unloading and loading times of planned dispatches and consider introducing empty dispatches, which can depart at any feasible time points (i.e., not only at time points associated with a planned dispatch). Empty dispatches can only occur for a predetermined amount of time after the unloading of a planned dispatch. Only a single empty dispatch is allowed between two consecutive planned dispatches (which is not restrictive if the triangular inequality holds). We allow only a single layover and a layover can only occur after a minimum number of hours of driving and on-duty time (a company policy).

4.4.2 Driver scheduling problem

Given a set of candidate time-feasible cycles, the transportation and operational costs for company and contractor, and the contractor's lane coverage, we solve an integer programming model to identify time-feasible driver schedules for four scenarios, each representing a different company resource planning strategy, listed in Table 4.2. The first scenario represents the most restrictive setting where the company only considers out-and-back cycles for its own drivers. In the second scenario, the company considers longer cycles, but still only for its own drivers. In the third scenario, the company considers outsourcing one-way moves to contractors. Finally, in the fourth, most flexible scenario, the company considers outsourcing one-way moves and cycles (in addition to using its own drivers).

Table 4.2: Resource planning strategies

Scenario	Resource type	Driver route type
Scenario 1	Company	Out-and-back
Scenario 2	Company	Cycle
Scenario 3	Company & Contractor	One-way (contractor) Cycle (company)
Scenario 4	Company & Contractor	One-way (contractor) Cycle (contractor) Cycle (company)

Table 4.3 describes the notation used in the IP model. The cost savings of a (company

or contractor) cycle $c \in C$ is the difference between the sum of the (unrealistic) base costs of the planned vehicle dispatches in the cycle and the sum of the (company or contractor) costs of performing the loaded and empty dispatches in the cycle with the minimum needed size vehicle. Similarly, the cost savings of a contractor one-way move for a planned vehicle dispatch ϑ is the difference between the unrealistic base cost of ϑ and the contractor one-way cost for performing it with the same vehicle type.

Table 4.3: Notation for driver scheduling problem

Sets	
C_{co}	Set of company cycles
C_{contr}	Set of contractor cycles
C	Set of cycles ($C = C_{co} \cup C_{contr}$)
Θ	Set of planned vehicle dispatches
Θ_{contr}	Set of planned vehicle dispatches contractor can execute as one-way ($\Theta_{contr} \subseteq \Theta$)
Parameters	
$\delta_{\vartheta c}$	1 if planned vehicle dispatch $\vartheta \in \Theta$ is in cycle $c \in C$, 0 otherwise
n_c	Number of planned vehicle dispatches in cycle $c \in C$
γ_c	Cost savings of company cycle $c \in C_{co}$
β_c	Cost savings of contractor cycle $c \in C_{contr}$
α_ϑ	Cost saving of contractor one-way move for dispatch $\vartheta \in \Theta_{contr}$
Decision variables	
x_ϑ	1 if planned vehicle dispatch $\vartheta \in \Theta_{contr}$ is outsourced as one-way move, 0 otherwise
y_c	1 if company cycle $c \in C_{co}$ is chosen, 0 otherwise
z_c	1 if contractor cycle $c \in C_{contr}$ is chosen, 0 otherwise
$u_{\vartheta c}$	1 if planned vehicle dispatch $\vartheta \in \Theta$ is covered by company cycle $c \in C_{co}$, 0 otherwise
$v_{\vartheta c}$	1 if planned vehicle dispatch $\vartheta \in \Theta$ is covered by contractor cycle $c \in C_{contr}$, 0 otherwise

$$\max \quad \sum_{\vartheta \in \Theta_{contr}} \alpha_\vartheta x_\vartheta + \sum_{c \in C_{co}} \gamma_c y_c + \sum_{c \in C_{contr}} \beta_c z_c \tag{4.1}$$

$$\text{subject to} \quad x_\vartheta + \sum_{c \in C_{co}} \delta_{\vartheta c} u_{\vartheta c} + \sum_{c \in C_{contr}} \delta_{\vartheta c} v_{\vartheta c} \leq 1 \quad \forall \vartheta \in \Theta \tag{4.2}$$

$$\sum_{\vartheta \in \Theta} \delta_{\vartheta c} u_{\vartheta c} = n_c y_c \quad \forall c \in C_{co} \tag{4.3}$$

$$\sum_{\vartheta \in \Theta} \delta_{\vartheta c} v_{\vartheta c} = n_c z_c \quad \forall c \in C_{contr} \tag{4.4}$$

$$x_\vartheta \in \{0, 1\} \quad \forall \vartheta \in \Theta_{contr} \tag{4.5}$$

$$y_c \in \{0, 1\} \quad \forall c \in C_{co} \tag{4.6}$$

$$z_c \in \{0, 1\} \quad \forall c \in C_{contr} \tag{4.7}$$

$$u_{\vartheta c} \in \{0, 1\} \quad \forall \vartheta \in \Theta, \quad \forall c \in C_{co} \tag{4.8}$$

$$v_{\vartheta c} \in \{0, 1\} \quad \forall \vartheta \in \Theta, \quad \forall c \in C_{contr} \tag{4.9}$$

Constraints (4.2) state that each dispatch can either be covered by one of the available options (i.e., a company cycle, contractor cycle or contractor one-way move) or remain uncovered. In case a dispatch is uncovered, it will be executed as out-and-back move with an empty return. Constraints (4.3) - (4.4) ensure that if a cycle (company or contractor) is chosen, then all planned dispatches in it will be covered by the same cycle. Finally, constraints (4.5) - (4.9) define the domain of the decision variables. The objective (depending on the chosen scenario) is to choose a set of company cycles, and contractor one-way moves and cycles if available, to maximize the cost savings over the (unrealistic) base scenario.

4.4.3 Out-and-backs

When a company only performs out-and-back cycles to execute dispatches, as is current practice at SF Express, we can formulate and efficiently solve the driver scheduling problem as a bipartite matching problem for each out-and-back lane pair (two lanes where one lane's destination is the other lane's origin) separately.

For a given out-and-back lane pair, two disjoint node sets in a bipartite graph represent the planned vehicle dispatches departing on opposite lanes and the edges between these nodes represent the time-feasible out-and-backs that can be formed between nodes (planned vehicle dispatches). The cost savings associated with each edge (matching) is calculated same as cost savings of any cycle explained in Section 4.4.2

4.5 Data generation

We designed a procedure to generate company and contractor costs per kilometer on each lane and for each vehicle type in South China,part of SF line-haul network. We considered 5 vehicles types at SF Express' disposal and we assumed each vehicle type is available on each lane for the line-haul operations. To represent a setting in which the cost per unit volume per unit distance decreases monotonically as the vehicle capacity increases, we assumed the average vehicle cost (per km) for company one-way moves listed in Table 4.4.

Table 4.4: Average vehicle cost - company one-way move

Vehicle type	1.5	3	5	7	14
Average cost per km ($)	6	6.2	6.7	7.4	7.8

4.5.1 Company one-way costs

Let μ_v represent the average one-way cost per km for vehicle type $v \in V$. For each lane $l \in L$, we generate one-way company cost per km, c_{lv}^{OW}, from a normal distribution with mean μ_v and the variance $\mu_v/8$. We choose a small variance relative to the distribution mean with the assumption that the company costs do not show high variability in company's line-haul network.

4.5.2 Company out-and-back costs

Company drivers expect their trips to start and end at their home base. Therefore, we consider the minimum viable trip for a company driver as single-loaded-leg out-and-back which consists of one loaded and one empty leg between the company driver's home base and a target terminal. As the drivers get paid for both loaded and unloaded kilometers they travel, the associated unrealistic total out-and-back cost per km for vehicle type v and lane l, c_{lv}^{ONB}, can be considered as

$$c_{lv}^{ONB} = (c_{lv}^{OW}\mu_l + c_{\hat{l}v}^{OW}\mu_{\hat{l}})/\mu_l \tag{4.10}$$

where $o_l = d_{\hat{l}}$, $d_l = o_{\hat{l}}$ and u_l is the length of lane l.

4.5.3 Contractor one-way costs

Companies would consider outsourcing one-way moves to contractors if the outsourced one-way move is cheaper than company single-loaded-leg out-and-back move. More specifically the one-way contractor cost (per km) for vehicle v on lane l, o_{lv}^{OW}, should not be more than the out-and-back company cost (per km) for vehicle v on lane l. Realistically, the one-way contractor costs (per km) should also be higher than one-way company costs (per kim). Otherwise, assuming companies manage the outsourcing risks (e.g. shortage of drivers) in the contracts, they would choose to outsource all of their line-haul transportation. To represent this relationship between company and contractor costs with different levels of negotiations, for each vehicle $v \in V$ on each lane $l \in L$, we generate one-way contractor costs (per km) from the uniform distributions whose boundaries are selected according to settings listed in Table 4.5. In Setting 1, contractor offers both low and high prices for different lanes in the line-haul network. In Setting 2, contractor offers low prices that is close to company costs for one-way moves while in Setting 3, the contractor prices are high.

The variability we introduce in one-way contractor costs might depend on the length of the contract and/or lane popularity. For instance, a contractor may offer lower price if the length of the contract is long and higher prices if it is short or a contractor might offer lower prices on unpopular lanes to attract potential shippers and high prices for lanes in high demand.

Table 4.5: Contractor one-way costs

	$o_{lv}^{OW} \sim \mathcal{U}(a,b)$	
Setting	a	b
Setting 1	c_{lv}^{OW}	c_{lv}^{ONB}
Setting 2	c_{lv}^{OW}	$(c_{lv}^{OW} + c_{lv}^{ONB})/2$
Setting 3	$(c_{lv}^{OW} + c_{lv}^{ONB})/2$	c_{lv}^{ONB}

4.5.4 Contractor cycle costs

Once contractors receive dispatches from multiple shippers, they also build round-trips to cover these dispatches that minimize the costs that are not covered by the outsourcing contracts. However, it is not always possible to construct cycles by combining dispatches from multiple shippers with low cost. The dispatches might be geographically dispersed and/or their departure/arrival times might not synchronized to let contractors build feasible driver schedules. For this reason, contractors may be inclined to offer discounts for cycled dispatches. For contractor cycle costs (per km), we consider four discount levels, 5%, 10%, 15% and 25%, over the contractor one-way costs (per km).

4.6 Computational Study

We tested our solution approach with a load plan generated for SF Express that includes 10,329 vehicle planned dispatches in South China for three days planning period (using the flow planning approach presented in Chapter 3) for five vehicle types in Table 4.4. We excluded any line-haul lanes which take more than 12 hours[1] and any commodities[2] that are on these lanes because they would never be served due to our choice of hours of service regulations (given in Section 4.3).

We choose a base model setting given in Table 4.6 for our experiments. All else being equal, we change one or a group of model parameters from the base model setting to investigate the impact of several modeling complexities and algorithmic choices on solution quality and speed for four scenarios presented in Table 4.2 and several negotiated price settings for outsourcing option presented in 4.5.

Table 4.6: Base model setting - driver scheduling problem

Modeling choices	
Operating hours at a terminal	7/24
Vehicle loading/unloading times	20 min / 20 min
Layover cost	50
Vehicle waiting time limit at a terminal	20 min
% of company lanes contractors operate	100%
# of empty dispatches between two consecutive planned dispatches	0
Minimum distance for layover	300 km
Minimum distance for outsourcing (one-ways and cycles)	50 km
Maximum cycle duration	48 h
Maximum number of layover in a cycle	1
Algorithmic choices	
Max number of planned dispatches in a cycle	4 (Scenarios 2,3,4), 2 (Scenario 1)
Minimum driving time before rest period	10
Minimum on-duty time before rest period	13
Time discretization (for empty dispatch creation)	60 min
Contractor price discount choice	
Setting	1
Cycle discount	10 %
Solver (Gurobi) settings	
Time limit	10 h
Relative MIP Optimality Gap	1e-4 (default value)

We used the computational cluster maintained by the School of Industrial and Systems Engineering at Georgia Institute of Technology to conduct our computational experiments. DFS for generating time-feasible n-cycles and optimization models were implemented in Python 2.7. The optimization models were solved by Gurobi 9.0. The computers used in the study had Xeon E5645 processors and 96 GB of RAM.

[1]Out of 3,422 lanes, there are 273 lanes whose travel time is more than 12 hours.

[2]Out of 60,073 commodities, there are 796 commodities whose direct paths take more than 12 hours.

Tables 4.7 - 4.13 show the experiment results on modeling choices and algorithmic ideas for four scenarios. With the time limit of 10 hours, the majority of the experiments finished with tiny gaps (less than 1%), while a few instance resulted with sizeable gaps (31% in the worst case).

In Scenario 1, when we only consider company out-and-backs, the cost saving is 6.53% and there are 1,261 dispatch pairs matched in an out-and-back cycle using the base model setting (Exp$\#$ 1) (Table 4.7). The cost saving goes up to 11.83% if we increase the waiting time limit at terminals from 20 minutes to an hour (Exp$\#$ 7). On the other hand, when company considers cycles up to 4 legs (Exp$\#$ 8) as in Scenario 2, the cost saving is 12.28%.

Among all scenarios using the base model setting (Exp$\#$ 1/8/30/54), we observe the largest cost savings (29.93%) in Scenarios 3 and 4. For each scenario, the experiments show that increase in loading and unloading times to 40 minutes at terminals decreases cost savings slightly compared to the base model setting (Exp$\#$ 2/9/31/55). Increasing the layover cost from \$50 to \$100 does not change the solution in terms of number of chosen cycles and/or one-ways. We observe that out of 9,829 cycles generated with up to 4 planned dispatches (without empty moves) (as shown in Figure 4.1), only 13 cycles make a layover. Out of 13, 8 cycles are chosen in each of the scenario with different layover costs.

Finally, as expected, we observe that the cost savings increase as the limit on maximum waiting time at a terminal increases from 20 minutes to 60 minutes as more time-feasible cycles are considered in the optimization model.

For Scenarios 2, 3 and 4, increasing the maximum number of planned dispatches in a cycle from 4 to 10 improves the cost saving while increasing the computational burden. Meanwhile, the number of chosen cycles for company (and contractor) decreases, which implies that the model chooses cycles with number of planned dispatches greater than 4. Table 4.17 summarizes the number of company and contractor cycles when the maximum number of planned dispatches is set to 10 and no empty moves are considered. We see that the model chooses cycles with up to 9 planned dispatches. No cycle with 10 planned

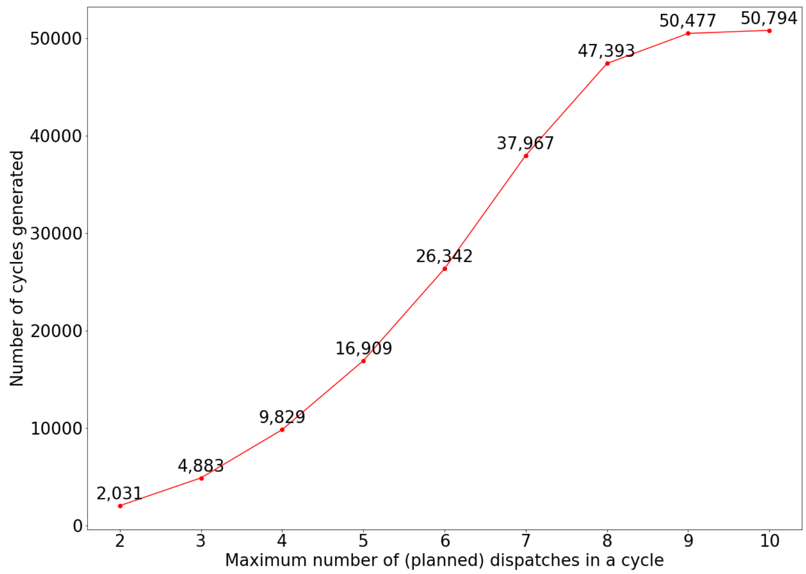


Figure 4.1: Cycle generation - no empty dispatches

dispatches are chosen in any of the three scenarios.

In Tables 4.9 - 4.13, decreasing minimum driving time before rest period from 10 hours to 8 hours (Exp# 24/48/72) slightly increases the cost savings, while decreasing minimum on-duty time from 13 hours to 12 hours (Exp# 27/51/75) does not change the cost savings.

In Tables 4.11 - 4.13, for Scenarios 3 and 4 respectively, decreasing the contractor line-haul coverage from 100% to 75% decreases the cost savings from 29.93% to 25.66% (Exp# 37/61). The results suggest that any limitation on contractor operations in the line-haul network would result in lost opportunity for the company.

In Table 4.14, we look at three settings we consider in Section 4.5 to generate contractor costs (per km) for one-way moves. For each setting, we investigate four levels of price discount (per km) on contractor cycles. With 10% price discount for contractor cycles, we observe few number of contractor cycles chosen in different experiments/scenarios. Considering that outsourcing cycles is not the main business model for contractors, seeing few contractor cycles is not surprising. When we consider a price negotiation which gives higher price discounts on the contractor cycles, we naturally see increase in the chosen contractor cycles, as reported in Table 4.14. The increase is sharper in Setting 2 compared to Setting 1, while in Setting 3, even the price discount 25$\%$ is not sufficient to make contractor cycles appealing. In Settings 1 and 2, we observe that increasing cycle discount from 5$\%$ to 25$\%$ slightly increases the cost savings.

In Table 4.15, we look into the impact of introducing empty moves between planned dispatches in a cycle under Scenario 4. The maximum number of planned dispatches in a cycle is 4. To keep computational study tractable, we only allow exploration of one empty dispatch within a predetermined waiting time after a planned dispatch arrival's unloading. When empty moves are introduced to cycles, cost savings improve between 1% to 5% for three waiting time limits. Table 4.16 show the details of chosen company and contractor cycles. We see that the model chooses long cycles for company drivers while the size and number of chosen contractor cycles remain small when we introduce empty moves between consecutive planned dispatches.

Finally, Table 4.18 shows the percentage of chosen company and contractor cycles with layover in Scenario 4 with the maximum number of planned dispatches in a cycle ranging from 2 to 10. We see that among the few (less than 10) chosen contractor cycles in each of these experiments, none of the contractor cycles have layovers. On the other hand, the percentage of chosen company cycles with layover increases from 0.16$\%$ to 3.17$\%$ as the maximum number of planned dispatches in a cycle increases from 2 to 10.

Table 4.7: Modeling Choices - scenario 1

Exp #		# Out-and-back matchings	Cost savings (%)	Run time (h)
	Vehicle unloading/loading times			
1	20 min/20 min	1,261	6.53%	0.02
2	40 min/40 min	1,224	6.30%	0.02
	Layover cost			
3	50	1,261	6.53%	0.02
4	100	1,261	6.52%	0.02
	Vehicle waiting time limit at a terminal			
5	20 min	1,261	6.53%	0.02
6	40 min	1,814	9.96%	0.02
7	60 min	2,132	11.83%	0.02

Table 4.8: Modeling Choices - scenario 2

Exp #		# Cycles		# One-ways	# Chosen cycles		# Chosen One-ways	Cost savings	Run time
		Co	Contr	Contr	Co	Contr	Contr	(%)	(h)
	Vehicle waiting time (min)								
8	20 min/20 min	9,829	0	0	1,519	0	0	12.28%	0.04
9	40 min/40 min	7,865	0	0	1,458	0	0	11.66%	0.03
	Layover cost								
10	50	9,829	0	0	1,519	0	0	12.28%	0.04
11	100	9,829	0	0	1,519	0	0	12.26%	0.05
	Vehicle waiting time limit at a terminal								
12	20 min	9,829	0	0	1,519	0	0	12.28%	0.04
13	40 min	46,600	0	0	1,857	0	0	19.55%	10.13
14	60 min	124,341	0	0	1,868	0	0	22.54%	10.37

Exp #		# Cycles		# One-ways	# Chosen cycles		# Chosen One-ways	Cost savings	Run time
		Co	Contr	Contr	Co	Contr	Contr	(%)	(h)
	Max number of planned dispatches in a cycle								
15	2	2,031	0	0	1,261	0	0	6.53%	0.02
16	3	4,883	0	0	1,512	0	0	10.26%	0.03
17	4	9,829	0	0	1,519	0	0	12.28%	0.04
18	5	16,909	0	0	1,484	0	0	13.56%	0.34
19	6	26,342	0	0	1,410	0	0	14.04%	10.08
20	7	37,967	0	0	1,336	0	0	14.13%	10.09
21	8	47,393	0	0	1,304	0	0	14.07%	10.12
22	9	50,477	0	0	1,335	0	0	14.44%	10.11
23	10	50,794	0	0	1,304	0	0	14.17%	10.12
	Minimum driving time before rest period								
24	8h	9,887	0	0	1,653	0	0	13.54%	0.04
25	10h	9,829	0	0	1,519	0	0	12.28%	0.04
26	12h	9,818	0	0	1,512	0	0	11.86%	0.05
	Minimum on-duty time before rest period								
27	12h	9,831	0	0	1,518	0	0	12.28%	0.04
28	13h	9,829	0	0	1,519	0	0	12.28%	0.04
29	15h	9,829	0	0	1,519	0	0	12.28%	0.04

Table 4.9: Algorithmic Choices - scenario 2

Table 4.10: Modeling Choices - scenario 3

Exp #		# Cycles		# One-ways	# Chosen cycles		# Chosen One-ways	Cost savings	Run time
		Co	Contr	Contr	Co	Contr	Contr	(%)	(h)
	Vehicle waiting time (min)								
30	20 min/20 min	9,829	0	6,391	1,482	0	4,171	29.93%	0.04
31	40 min/40 min	7,865	0	6,391	1,439	0	4,326	29.38%	0.04
	Layover cost								
32	50	9,829	0	6,391	1,482	0	4,171	29.93%	0.04
33	100	9,829	0	6,391	1,482	0	4,171	30.03%	0.04
	Vehicle waiting time limit at a terminal								
34	20 min	9,829	0	6,391	1,482	0	4,171	29.93%	0.04
35	40 min	46,600	0	6,391	2,007	0	3,062	34.62%	10.11
36	60 min	124,341	0	6,391	1,987	0	2,724	36.28%	10.30
	% of company lanes contractors operate								
37	75%	9,829	0	4,693	1,489	0	3,110	25.66%	0.04
38	100%	9,829	0	6,391	1,482	0	4,171	29.93%	0.04

Table 4.11: Algorithmic Choices - scenario 3

Exp #		# Cycles		# One-ways	# Chosen cycles		# Chosen One-ways	Cost savings	Run time
		Co	Contr	Contr	Co	Contr	Contr	(%)	(h)
	Max number of planned dispatches in a cycle								
39	2	2,031	0	6,391	1,252	0	5,220	26.66%	0.02
40	3	4,883	0	6,391	1,486	0	4,458	28.74%	0.02
41	4	9,829	0	6,391	1,482	0	4,171	29.93%	0.04
42	5	16,909	0	6,391	1,455	0	4,060	30.54%	0.23
43	6	26,342	0	6,391	1,431	0	4,038	30.94%	0.49
44	7	37,967	0	6,391	1,419	0	4,002	31.15%	5.61
45	8	47,393	0	6,391	1,391	0	4,007	31.23%	10.12
46	9	50,477	0	6,391	1,390	0	4,005	31.26%	10.13
47	10	50,794	0	6,391	1,359	0	4,026	31.17%	10.14
	Minimum driving time before rest period								
48	8h	9,887	0	6,391	1,494	0	4,133	30.54%	0.04
49	10h	9,829	0	6,391	1,482	0	4,171	29.93%	0.04
50	12h	9,818	0	6,391	1,476	0	4,191	29.69%	0.04
	Minimum on-duty time before rest period								
51	12h	9,831	0	6,391	1,482	0	4,171	29.93%	0.04
52	13h	9,829	0	6,391	1,482	0	4,171	29.93%	0.04
53	15h	9,829	0	6,391	1,482	0	4,171	29.93%	0.04

Table 4.12: Modeling Choices - scenario 4

Exp #		# Cycles		# One-ways	# Chosen cycles		# Chosen One-ways	Cost savings	Run time
		Co	Contr	Contr	Co	Contr	Contr	(%)	(h)
	Vehicle waiting time (min)								
54	20 min/20 min	9,783	46	6,391	1,477	7	4,172	29.93%	0.04
55	40 min/40 min	7,831	34	6,391	1,436	3	4,326	29.38%	0.03
	Layover cost								
56	$50	9,783	46	6,391	1,477	7	4,172	29.93%	0.04
57	$100	9,783	46	6,391	1,477	7	4,172	30.03%	0.04
	Vehicle waiting time limit at a terminal								
58	20 min	9,783	46	6,391	1,477	7	4,172	29.93%	0.04
59	40 min	46,490	110	6,391	1,984	10	3,077	34.63%	10.21
60	60 min	124,092	249	6,391	1,940	10	2,752	36.08%	10.38
	% of company lanes contractors operate								
61	75%	9,783	46	4,693	1,490	7	3,102	25.66%	0.05
62	100%	9,783	46	6,391	1,477	7	4,172	29.93%	0.04

Table 4.13: Algorithmic Choices - scenario 4

Exp #		# Cycles		# One-ways	# Chosen cycles		# Chosen One-ways	Cost savings	Run time
		Co	Contr	Contr	Co	Contr	Contr	(%)	(h)
	Max number of planned dispatches in a cycle								
63	2	2,019	12	6,391	1,246	6	5,220	26.66%	0.02
64	3	4,840	43	6,391	1,477	9	4,458	28.74%	0.03
65	4	9,783	46	6,391	1,477	7	4,172	29.93%	0.04
66	5	16,861	48	6,391	1,453	5	4,066	30.54%	0.23
67	6	26,294	48	6,391	1,424	6	4,038	30.94%	0.40
68	7	37,919	48	6,391	1,404	3	4,005	31.15%	3.54
69	8	47,345	48	6,391	1,389	3	4,000	31.23%	10.12
70	9	50,429	48	6,391	1,390	4	4,013	31.26%	10.15
71	10	50,746	48	6,391	1,386	3	4,013	31.26%	10.12
	Minimum driving time before rest period								
72	8h	9,841	46	6,391	1,491	7	4,134	30.54%	0.04
73	10h	9,783	46	6,391	1,477	7	4,172	29.93%	0.04
74	12h	9,772	46	6,391	1,472	7	4,193	29.69%	0.03
	Minimum on-duty time before rest period								
75	12h	9,785	46	6,391	1,477	7	4,171	29.93%	0.04
76	13h	9,783	46	6,391	1,477	7	4,172	29.93%	0.04
77	15h	9,783	46	6,391	1,477	7	4,172	29.93%	0.04

Table 4.14: Contractor discount analysis - scenario 4

Exp #		# Cycles		# One-ways	# Chosen cycles		# Chosen One-ways	Cost savings	Run time
		Co	Contr	Contr	Co	Contr	Contr	(%)	(h)
	Setting 1								
	Cycle Discount								
78	5 %	9,827	2	6,391	1,482	0	4,171	29.93%	0.04
79	10 %	9,783	46	6,391	1,477	7	4,172	29.93%	0.03
80	15 %	9,631	198	6,391	1,460	27	4,166	29.93%	0.04
81	25 %	8,536	1,293	6,391	1,344	150	4,155	30.02%	0.04
	Setting 2								
	Cycle Discount								
82	5 %	9,767	62	6,391	1,470	6	4,316	38.15%	0.03
83	10 %	9,341	488	6,391	1,401	80	4,307	38.17%	0.04
84	15 %	8,298	1,531	6,391	1,218	268	4,301	38.24%	0.04
85	25 %	2,767	7,062	6,391	501	1,009	4,228	38.95%	0.04
	Setting 3								
	Cycle Discount								
86	5 %	9,829	0	6,391	1,513	0	4,033	21.47%	0.04
87	10 %	9,829	0	6,391	1,513	0	4,033	21.47%	0.05
88	15 %	9,829	0	6,391	1,513	0	4,033	21.47%	0.05
89	25 %	9,829	0	6,391	1,513	0	4,033	21.47%	0.04

Table 4.15: Empty move analysis - scenario 4

Exp #		# Cycles		# One-ways	# Chosen cycles		# Chosen One-ways	Cost savings	Run time
		Co	Contr	Contr	Co	Contr	Contr	(%)	(h)
	Without empty dispatches between consecutive planned dispatches								
	Vehicle waiting time								
90	0 min	117	2	6,391	95	1	6,297	22.87%	0.02
91	10 min	2,563	21	6,391	896	5	5,241	26.31%	0.03
92	20 min	9,783	46	6,391	1,477	7	4,172	29.93%	0.04
	With empty dispatches between consecutive planned dispatches								
	Vehicle waiting time								
93	0 min	23,365	58	6,391	582	5	5,766	23.71%	0.07
94	10 min	1,844,063	1,653	6,391	1,996	9	3,449	31.41%	14.35
95	20 min	7,987,990	6,969	6,391	2,058	9	2,838	34.68%	26.95

Table 4.16: Empty move analysis - number of chosen company and contractor cycles

	Without empty dispatches						With empty dispatches					
	0 min		10 min		20 min		0 min		10 min		20 min	
(n-cycle, # empty dispatches)	Co	Contr	Co	Contr	Co	Contr	Co	Contr	Co	Contr	Co	Contr
(2,0)	87	1	544	3	754	4	80	0	444	3	543	2
(3,0)	6	0	226	2	380	3	3	0	122	2	192	2
(3,1)	0	0	0	0	0	0	304	5	554	4	411	5
(4,0)	2	0	126	0	343	0	0	0	44	0	104	0
(4,1)	0	0	0	0	0	0	33	0	234	0	258	0
(4,2)	0	0	0	0	0	0	122	0	153	0	77	0
(5,1)	0	0	0	0	0	0	1	0	137	0	193	0
(5,2)	0	0	0	0	0	0	29	0	111	0	96	0
(6,2)	0	0	0	0	0	0	4	0	109	0	120	0
(6,3)	0	0	0	0	0	0	3	0	23	0	16	0
(7,3)	0	0	0	0	0	0	3	0	50	0	0	0
(8,4)	0	0	0	0	0	0	0	0	15	0	3	0

Table 4.17: Number of chosen company and contractor cycles

	2-cycle		3-cycle		4-cycle		5-cycle		6-cycle		7-cycle		8-cycle		9-cycle		10-cycle	
	Co	Contr	Co	Contr	Co	Contr	Co	Contr	Co	Contr	Co	Contr	Co	Contr	Co	Contr	Co	Contr
Exp #																		
23	682	0	291	0	178	0	73	0	40	0	25	0	11	0	4	0	0	0
47	692	0	296	0	191	0	86	0	40	0	35	0	16	0	3	0	0	0
71	696	2	301	1	199	0	89	0	48	0	35	0	16	0	2	0	0	0

Table 4.18: Chosen cycles with layover - scenario 4 - without empty dispatches

Exp #	# cycle legs	% layovers	
		Co	Contr
63	2	0.16%	0.00%
64	3	0.27%	0.00%
65	4	0.54%	0.00%
66	5	1.51%	0.00%
67	6	2.25%	0.00%
68	7	2.78%	0.00%
69	8	3.02%	0.00%
70	9	3.17%	0.00%
71	10	3.17%	0.00%

4.7 Conclusion

In this chapter, we have focused on driver scheduling and we have analyzed the value of outsourcing transportation for different negotiated prices with contractors through an extensive computational study.

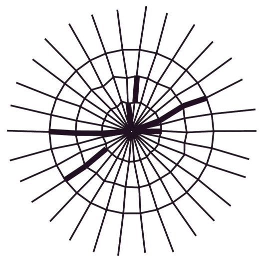
Our approach generates a set of time-feasible candidate cycles of chosen length using a depth-first search algorithm, which considers a representative set of hours of service regulations (e.g. maximum driving time), terminal operations (e.g. loading and unloading) and company policies (e.g. maximum number of layovers in a cycle). We solve an integer programming model that identifies a subset of company cycles (and contractor cycles and one-way moves if the outsourcing option is available) that maximizes the cost savings over the (unrealistic) scenario in which company drivers perform one-way moves and return empty. When a company only performs out-and-back cycles, as is current practice at SF Express, we efficiently choose the set of cycles by solving a bipartite matching problem for each out-and-back lane pair separately.

A natural next investigation is to try and integrate flow planning (Chapter 3) and driver scheduling (Chapter 4). Such a holistic approach likely reduces total costs, but also significantly increases the complexity of the optimization problem.

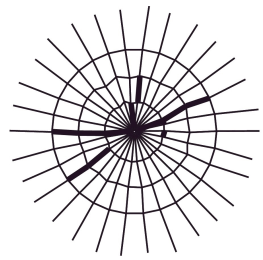
Appendices

APPENDIX A

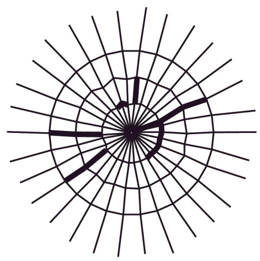
PROACTIVE ROUTE GUIDANCE TO AVOID CONGESTION



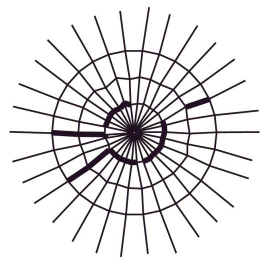
(a) $\tau = 0\%, \rho^*_\tau = 3.05$



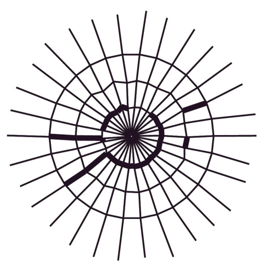
(b) $\tau = 5\%, \rho^*_\tau = 2.46$



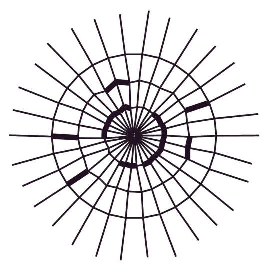
(c) $\tau = 10\%, \rho^*_\tau = 1.79$



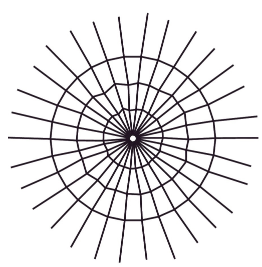
(d) $\tau = 15\%, \rho^*_\tau = 1.35$



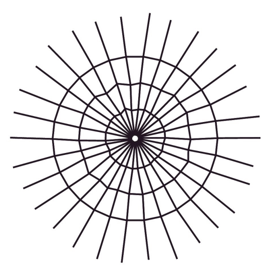
(e) $\tau = 20\%, \rho^*_\tau = 1.14$



(f) $\tau = 25\%, \rho^*_\tau = 1.02$

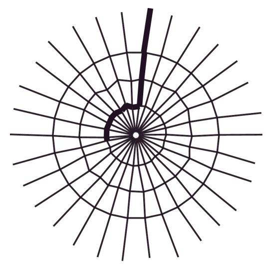


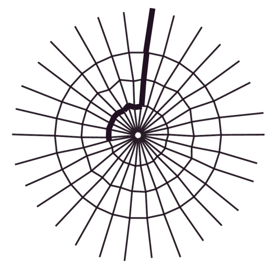
(g) $\tau = 30\%, \rho^*_\tau = 0.89$

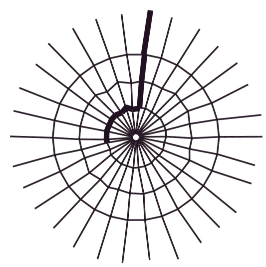


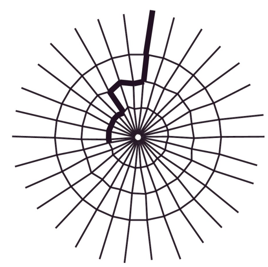
(h) $\tau = 35\%, \rho^*_\tau = 0.81$

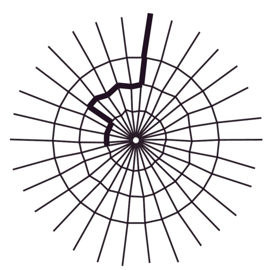
Figure A.1: Congestion for different values of τ

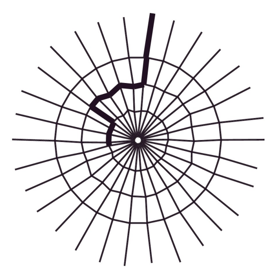

(a) $\tau = 0\%$

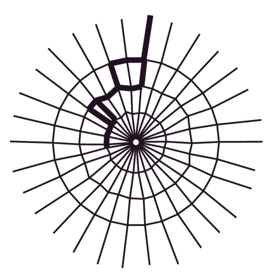

(b) $\tau = 5\%$


(c) $\tau = 10\%$


(d) $\tau = 15\%$


(e) $\tau = 20\%$


(f) $\tau = 25\%$


(g) $\tau = 30\%$

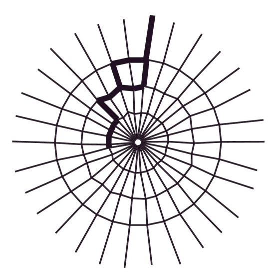

(h) $\tau = 35\%$

Figure A.2: Selected paths for a single OD pair for different values of τ

REFERENCES

[1] S. David, E. Bill, and L. Tim, "2019 urban mobiity report," Tech. Rep., 2019.

[2] *Coronavirus disease (COVID-19) - Situation Report – 104*, `https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200503-covid-19-sitrep-104.pdf?sfvrsn=53328f46_2`, Accessed: 2020-05-03.

[3] *Didi, SF Express, and JD.com to maintain operations in quarantined Wuhan*, `https://kr-asia.com/didi-sf-express-and-jd-com-to-maintain-operations-in-quarantined-wuhan`, Accessed: 2020-05-10.

[4] *Coronavirus lockdown sparks expansion of drones and robot deliveries*, `https://asia.nikkei.com/Opinion/Coronavirus-lockdown-sparks-expansion-of-drones-and-robot-deliveries`, Accessed: 2020-05-03.

[5] N. Boland, M. Hewitt, L. Marshall, and M. Savelsbergh, "The continuous-time service network design problem," *Operations Research*, vol. 65, no. 5, pp. 1303–1321, 2017.

[6] J. R. Nocera, I. Arsik, P. Keskinocak, A. Lepley-Flood, J. J. Lah, A. I. Levey, and G. J. Esper, "The feasibility of measuring gait in an outpatient cognitive neurology clinical setting," *Journal of Alzheimer's Disease*, vol. 71, no. s1, s51–s55, 2019.

[7] I. Arsik, J. Frediani, D. Frezza, W. Chen, T. Ayer, P. Keskinocak, R. Jin, J. Konomi, S. Barlow, S. Xanthakos, and et al., "Alanine aminotransferase as a monitoring biomarker in children with nonalcoholic fatty liver disease: A secondary analysis using tonic trial data," *Children*, vol. 5, no. 6, p. 64, 2018.

[8] J. E. Lavine, J. B. Schwimmer, M. L. Van Natta, J. P. Molleston, K. F. Murray, P. Rosenthal, S. H. Abrams, A. O. Scheimann, A. J. Sanyal, N. Chalasani, *et al.*, "Effect of vitamin E or metformin for treatment of nonalcoholic fatty liver disease in children and adolescents: The TONIC randomized controlled trial," *Jama*, vol. 305, no. 16, pp. 1659–1668, 2011.

[9] H. S. Mahmassani and S. Peeta, *Network Performance Under System Optimal and User Equilibrium Dynamic Assignments: Implications for Advanced Traveler Information Systems*. 1993.

[10] L. S. Ben-Akiva M.E., *Discrete Choice Analysis: Theory and Application to Travel Demand*. MIT press, 1985.

[11] M. Florian and D. Hearn, "Network equilibrium and pricing," in *Handbook of Transportation Science*. Springer US, 2003, pp. 373–411.

[12] Y. Sheffi, *Urban Transportation Network.Equilibrium Analysis with Mathematical Programming Methods*. Prentice Hall, 1985.

[13] J. de Dios Ortuzar and L. G. Willumsen, *Modelling Transport*. John Wiley & Sons, 2011.

[14] D. K. Merchant and G. L. Nemhauser, "A model and an algorithm for the dynamic traffic assignment problems," *Transportation Science*, vol. 12, no. 3, pp. 183–199, 1978.

[15] M. Papageorgiou, "Dynamic modeling, assignment, and route guidance in traffic networks," *Transportation Research Part B: Methodological*, vol. 24, no. 6, pp. 471–495, 1990.

[16] S. Peeta and A. K. Ziliaskopoulos, "Foundations of dynamic traffic assignment: The past, the present and the future," *Networks and Spatial Economics*, vol. 1, no. 3, pp. 233–265, 2001.

[17] M. E. Ben-Akiva, S. Gao, Z. Wei, and Y. Wen, "A dynamic traffic assignment model for highly congested urban networks," *Transportation Research Part C: Emerging Technologies*, vol. 24, pp. 62–82, 2012.

[18] O. Jahn, R. H. Möhring, A. S. Schulz, and N. E. Stier-Moses, "System-optimal routing of traffic flows with user constraints in networks with congestion," *Operations Research*, vol. 53, no. 4, pp. 600–616, 2005.

[19] S. Sen, R. Pillai, S. Joshi, and A. K. Rathi, "A mean-variance model for route guidance in advanced traveler information systems," *Transportation Science*, vol. 35, no. 1, pp. 37–49, 2001.

[20] M. Lujak, S. Giordani, and S. Ossowski, "Route guidance: Bridging system and user optimization in traffic assignment," *Neurocomputing*, vol. 151, pp. 449–460, 2015.

[21] K. Bartlett, J. Lee, S. Ahmed, G. Nemhauser, J. Sokol, and B. Na, "Congestion-aware dynamic routing in automated material handling systems," *Computers and Industrial Engineering*, vol. 70, pp. 176–182, 2014.

[22] M. Kaspi and J. M. A. Tanchoco, "Optimal flow path design of unidirectional agv systems," *International Journal of Production Research*, vol. 28, no. 6, pp. 1023–1030, 1990.

[23] J. Y. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. 712–716, 1971.

[24] D. Eppstein, "Finding the k shortest paths," *SIAM Journal on Computing*, vol. 28, no. 2, pp. 652–673, 1998.

[25] A. P. Punnen and R. Zhang, *Bottleneck flows in networks*, 2007. arXiv: `0712.3858 [cs.DS]`.

[26] J. R. Correa, A. S. Schulz, and N. E. Stier-Moses, "Fast, fair, and efficient flows in networks," *Operations Research*, vol. 55, no. 2, pp. 215–225, 2007.

[27] V. Bayram, "Optimization models for large scale network evacuation planning and management: A literature review," *Surveys in Operations Research and Management Science*, vol. 21, no. 2, pp. 63–84, 2016.

[28] F. D. Ford L.R., *Maximal flow through a network*. Can. j. Math., 1956, pp. 399–404.

[29] C. T. Gabriel, "Service network design in freight transportation," *European Journal of Operational Research*, vol. 122, no. 2, 272–288, 2000.

[30] W. B. Powell and Y. Sheffi, "The load planning problem of motor carriers: Problem description and a proposed solution approach," *Transportation Research Part A: General*, vol. 17, no. 6, pp. 471–480, 1983.

[31] W. B. Powell, "A local improvement heuristic for the design of less-than-truckload motor carrier networks," *Transportation Science*, vol. 20, no. 4, pp. 246–257, 1986.

[32] W. B. Powell and Y. Sheffi, "Or practice - design and implementation of an interactive optimization system for network design in the motor carrier industry," *Operations Research*, vol. 37, no. 1, pp. 12–29, 1989.

[33] W. B. Powell and I. A. Koskosidis, "Shipment routing algorithms with tree constraints," *Transportation Science*, vol. 26, no. 3, pp. 230–245, 1992.

[34] A. I. Jarrah, E. Johnson, and L. C. Neubert, "Large-scale, less-than-truckload service network design," *Operations Research*, vol. 57, no. 3, pp. 609–625, 2009.

[35] A. Erera, M. Hewitt, M. Savelsbergh, and Y. Zhang, "Creating schedules and computing operating costs for LTL load plans," *Computers & Operations Research*, vol. 40, no. 3, pp. 691–702, 2013.

[36] K. Lindsey, A. Erera, and M. Savelsbergh, “Improved integer programming-based neighborhood search for less-than-truckload load plan design,” *Transportation Science*, vol. 50, no. 4, pp. 1360–1379, 2016.

[37] J. Medina, M. Hewitt, F. Lehuédé, and O. Péton, “Integrating long-haul and local transportation planning: The service network design and routing problem,” *EURO Journal on Transportation and Logistics*, vol. 8, no. 2, pp. 119–145, 2019.

[38] C. Barnhart and R. R. Schneur, “Air network design for express shipment service,” *Operations Research*, vol. 44, no. 6, pp. 852–863, 1996.

[39] D. Kim, C. Barnhart, K. Ware, and G. Reinhardt, “Multimodal express package delivery: A service network design application,” *Transportation Science*, vol. 33, no. 4, pp. 391–407, 1999.

[40] T. Grünert and H.-J. Sebastian, “Planning models for long-haul operations of postal and express shipment companies,” *European Journal of Operational Research*, vol. 122, no. 2, pp. 289–309, 2000.

[41] C. Barnhart, N. Krishnan, D. Kim, and K. Ware, “Network design for express shipment delivery,” *Computational Optimization and Applications*, vol. 21, no. 3, pp. 239–262, 2002.

[42] H. N. Newton, C. Barnhart, and P. H. Vance, “Constructing railroad blocking plans to minimize handling costs,” *Transportation Science*, vol. 32, no. 4, pp. 330–345, 1998.

[43] C. Barnhart, H. Jin, and P. H. Vance, “Railroad blocking: A network design application,” *Operations Research*, vol. 48, no. 4, pp. 603–614, 2000.

[44] E. Zhu, T. G. Crainic, and M. Gendreau, “Scheduled service network design for freight rail transportation,” *Operations Research*, vol. 62, no. 2, pp. 383–400, 2014.

[45] C. Chen, T. Dollevoet, and J. Zhao, “One-block train formation in large-scale railway networks: An exact model and a tree-based decomposition algorithm,” *Transportation Research Part B: Methodological*, vol. 118, pp. 1–30, 2018.

[46] “Multicommodity, multimode freight transportation: A general modeling and algorithmic framework for the service network design problem,” *Transportation Research Part B: Methodological*, vol. 20, no. 3, pp. 225–242, 1986.

[47] “Operational planning of a large-scale multi-modal transportation system,” *European Journal of Operational Research*, vol. 156, no. 1, pp. 41–53, 2004.

[48] T. Yamada, B. F. Russ, J. Castro, and E. Taniguchi, “Designing multimodal freight transport networks: A heuristic approach and applications,” *Transportation Science*, vol. 43, no. 2, pp. 129–143, 2009.

[49] T. G. Crainic, M. Gendreau, and J. M. Farvolden, “A simplex-based tabu search method for capacitated network design,” *INFORMS Journal on Computing*, vol. 12, no. 3, pp. 223–236, 2000.

[50] T. G. Crainic and M. Gendreau, “Cooperative parallel tabu search for capacitated network design,” *Journal of Heuristics*, vol. 8, no. 6, pp. 601–627, 2002.

[51] I. Ghamlouche, T. G. Crainic, and M. Gendreau, “Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design,” *Operations Research*, vol. 51, no. 4, pp. 655–667, 2003.

[52] I. Ghamlouche, T. G. Crainic, and M. Gendreau, “Path relinking, cycle-based neighbourhoods and capacitated multicommodity network design,” *Annals of Operations Research*, vol. 131, no. 1, pp. 109–133, 2004.

[53] T. L. Magnanti and R. T. Wong, “Network design and transportation planning: Models and algorithms,” *Transportation Science*, vol. 18, no. 1, pp. 1–55, 1984.

[54] W. Nicole, “Service network design for freight transportation: A review,” *OR Spectrum*, vol. 30, no. 1, 77–112, 2008.

[55] J. M. Farvolden and W. B. Powell, “Subgradient methods for the service network design problem,” *Transportation Science*, vol. 28, no. 3, pp. 256–272, 1994.

[56] A. Erera, M. Hewitt, M. Savelsbergh, and Y. Zhang, “Improved load plan design through integer programming based local search,” *Transportation Science*, vol. 47, no. 3, pp. 412–427, 2013.

[57] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[58] M. Savelsbergh and M. Sol, “Drive: Dynamic routing of independent vehicles,” *Operations Research*, vol. 46, no. 4, pp. 474–490, 1998.

[59] Y. Rochat and F. Semet, “A tabu search approach for delivering pet food and flour in Switzerland,” *Journal of the Operational Research Society*, vol. 45, no. 11, pp. 1233–1246, 1994.

[60] G. Desaulniers, J. Desrosiers, A. Lasry, and M. M. Solomon, “Crew pairing for a regional carrier,” in *Computer-aided transit scheduling*, 1999, pp. 19–41.

[61] A. M. Campbell and M. Savelsbergh, "Efficient insertion heuristics for vehicle routing and scheduling problems," *Transportation science*, vol. 38, no. 3, pp. 369–378, 2004.

[62] Federal Motor Carrier Safety Administration (2011) Hours of service of drivers. Federal Register, vol. 76, no. 248, pp. 81134–81188.

[63] Federal Motor Carrier Safety Administration (2008) Hours of service of drivers. Federal Register, vol. 73, no. 224, pp. 69569–69586.

[64] H. Xu, Z.-L. Chen, S. Rajagopal, and S. Arunapuram, "Solving a practical pickup and delivery problem," *Transportation Science*, vol. 37, no. 3, pp. 347–364, 2003.

[65] C. Archetti and M. Savelsbergh, "The trip scheduling problem," *Transportation Science*, vol. 43, no. 4, pp. 417–431, 2009.

[66] A. Goel and L. Kok, "Truck driver scheduling in the United States," *Transportation science*, vol. 46, no. 3, pp. 317–326, 2012.

[67] M.-E. Rancourt, J.-F. Cordeau, and G. Laporte, "Long-haul vehicle routing and scheduling with working hour rules," *Transportation Science*, vol. 47, no. 1, pp. 81–107, 2013.

[68] A. Goel and V. Gruhn, "Drivers' working hours in vehicle routing and scheduling," in *2006 IEEE Intelligent Transportation Systems Conference*, 2006, pp. 1280–1285.

[69] A. Goel, "Truck driver scheduling in the European Union," *Transportation Science*, vol. 44, no. 4, pp. 429–441, 2010.

[70] A. L. Kok, E. W. Hans, and J. M. J. Schutten, "Optimizing departure times in vehicle routes," *European Journal of Operational Research*, vol. 210, no. 3, pp. 579–587, 2011.

[71] A. Goel and L. M. Rousseau, "Truck driver scheduling in Canada," *Journal of Scheduling*, vol. 15, no. 6, pp. 783–799, 2012.

[72] A. Goel, C. Archetti, and M. Savelsbergh, "Truck driver scheduling in Australia," *Computers & Operations Research*, vol. 39, no. 5, pp. 1122–1132, 2012.

[73] G. Zäpfel and M. Bögl, "Multi-period vehicle routing and crew scheduling with outsourcing options," *International Journal of Production Economics*, vol. 113, no. 2, pp. 980–996, 2008.